TU Clausthal

Technische Universität Clausthal

Institute for Software and Systems Engineering

Master Thesis

Towards Decentralized Data Marketplaces

Priyanka Sharma

Matr.-Nr.: 476454

05.06.2019

First assessor: Prof. Dr. Andreas Rausch

Second assessor: Prof. Dr. Marco Kuhrmann

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig angefertigt und alle Quellen und Hilfsmittel vollständig angegeben habe.

Hiermit erkläre ich mich einverstanden, dass meine Masterarbeit in der Instituts- und Universitätsbibliothek ausgelegt und zur Einsichtnahme aufbewahrt werden darf.

Clausthal-Zellerfeld, den 05.06.2019 _____

(Priyanka Sharma)

Abstract

Data is the oil of the 21st century. In the past years the awareness about the benefits of data and its usage has increased. A growing number of sectors recognize the benefits of data and are collecting data. Data is also the fuel for Artificial Intelligence.

Although today more data is generated and collected than ever before, there is a gap in its usage and acquisition. On one hand there are data collectors who collect data from data producers such as sensors, machines, users and so on. And on the other hand, there is a vast variety of data consumers such as researchers, educational institutions, artificial intelligence models, start-ups etc. who need data. In many cases for better decision making, improving efficiency and consumer experience even data collectors require other data which outside the internal boundaries. This had led many firms to look outside their boundaries and use commercialization mechanisms such as data brokers.

To tackle the increasing demand of data, a common data sharing platform is required. A data marketplace would serve as a platform where the data producers can sell data and the others can consume it. Data marketplaces have multiple centralized and decentralized approaches.

In order to establish direct data exchange between data producers and data consumers this thesis explores how a decentralized data marketplace can be designed. Furthermore, a real-world example where a data marketplace can be established, and a system architecture and proof of concept prototype is showed.

Table of Contents

List of Figures

List of tables

I. Glossary

Bitcoin: Bitcoin is a cryptocurrency. It is a decentralized digital currency without a central bank or single administrator that can be sent from user to user on the peer-to-peer bitcoin network.

Blockchain: Blockchain at its core is a distributed and decentralized open ledger that is cryptographically managed and updated various consensus protocols and agreements among its peers.

Buyers: Data consumers, or users who want to buy data from the data marketplace.

Consensus: Consensus is a process by with distributed systems reach an agreement amongst the nodes.

Data: Information, especially facts or numbers, collected to be examined and considered and used to help decision-making, or information in an electronic form that can be stored and used by a computer.

Data marketplace: A data marketplace is a platform where users buy and sell data.

Data Exchange: Data exchange is a process of data being shared between various parties or computers.

Data broker: Data brokers are entities that collect information about consumers, and then sell that data to other data brokers, companies, and/or individuals.

Decentralized system: A decentralized system is a system in which there is no central point where the decision is made.

Ethereum: Ethereum is a blockchain-based system with special scripting functionality that allows other developers to build decentralized and distributed applications on top of it.

Recycling: Recycling is the process of converting waste materials into new materials and objects.

Sellers: Data producers, providers or users who want to sell their data on the data marketplace.

Smart contracts: In Ethereum smart contracts are programs that run on the Ethereum computing infrastructure.

Solidity: A procedural programming language with syntax that is like JavaScript. C++ or Java.

Web 3: The third version of web, first proposed by Dr. Gavin Wood which represents which focuses on decentralized applications built on decentralized protocols.

1. Introduction

1.1 Motivation

Data is the new oil [1] *.*

Oil changed our world and the economy. Oil, as a technology fueled an economic force. Today, data is fueling a very similar change [1]. It is just not affecting high-tech sectors but also revolutionizing low-tech companies and sectors around the world. The new oil clearly shows that many companies and industries are shifting towards a data-driven strategy [1]. The events of the last 20 years have fundamentally changed the way data is treated. [2]. And this data is not some waste product but buried treasure waiting to be discovered. Data helps decision makers to take better decisions and provide better consumer experience. It also helps in making strategies and develop appropriate analytical techniques [3]. And most importantly data is the foundation for AI and machine learning applications. Machine learning and AI algorithms learn from data.

.

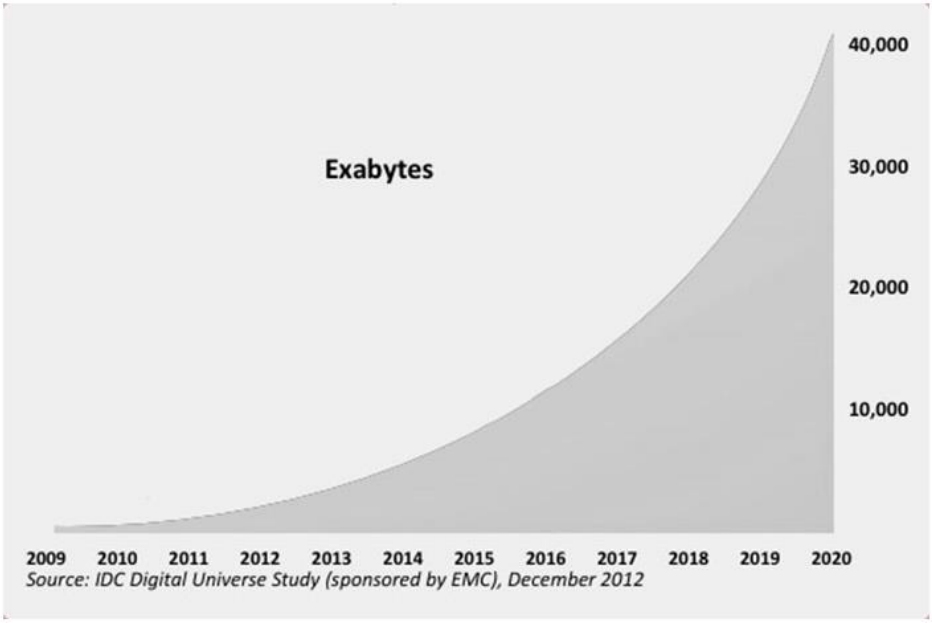

Figure 1 The amount of data in the world.

(Source: Arent Van't Spijker [1] *)*

Figure 1, shows the amount of data generated from 2009 and 2020. The amount of data generated every year increases drastically and it has been forecasted that the amount of data generated in the world will be 40,000 exabytes.

It's just not software-gaits or big high-tech companies but today many small industries including non-conventional low-tech sectors are also collecting data to leverage the value of data. But despite the need for data and its increasing collection, it is not easy to find relevant data easily. On one hand there are data producers such as individuals, companies, machines, sensors, government etc. On the other hand, there are data consumers such as AI applications, researchers, educational institutes, various industries and many other enterprises who are not able to produce data but need data.

The data acquisition problem.

Despite the increasing need of data and it's growing amount, data is not being shared or traded openly and transparently on a large scale [4] . Currently data is mostly acquired from data brokers, Data brokers—companies that collect consumers' personal information and resell or share that information with others—are important participants in this Big Data economy [5]. The lack of transparency of data brokers has raised many questions against their trustworthiness [5]. In addition to data brokers one-to-one business contracts for data

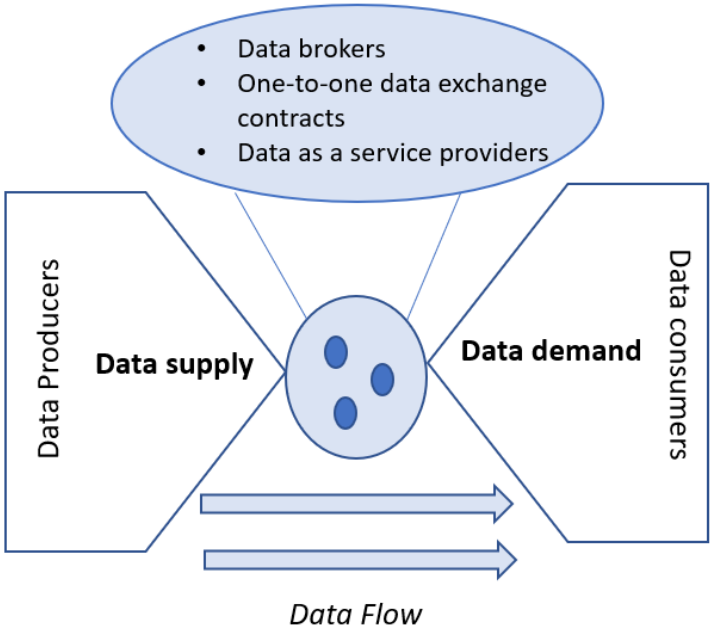

Figure 2 Current major data exchange mechanisms

(Source: Own analysis)

exchange are an alternative solution for data acquisition but these is less efficient as every actor needs contact each other directly and reach an agreement [6]. There are also many emerging platforms providing data as a service.

Data marketplaces are the new business models.

Thus, in today's data economy a common data sharing platform is required. Data marketplaces serve a valuable role in simplifying the data discovery process. With the help of a data marketplaces data producers can sell data and the others can consume it.

Conceptually, data marketplaces are multi-sided platforms, where a digital intermediary connects data producers, data consumers, and other complementary technology providers. These platforms would, generate value for both data buyers and sellers through enhanced market efficiency, resource allocation efficiency, and an improved match between supply and demand [4]. Understanding these data requirements a few data marketplaces have been established such as Infochimps[1], Factual[2], Azure[3], and DataMarket[4] have been established in the past years. But most of these are privately owned data marketplaces such as e-commerce online giants such as Amazon[5] or eBay[6]. The basic model of privately-owned data marketplaces is very similar to the working model of many privately-owned e-commerce platforms.

[1] www.infochimps.com
[2] www.factual.com
[3] www.datamarket.azure.com
[4] www.datamarket.com
[5] www.amazon.com
[6] www.ebay.com

Figure 3. shows a basic model of a privately-owned data marketplace. A company or group

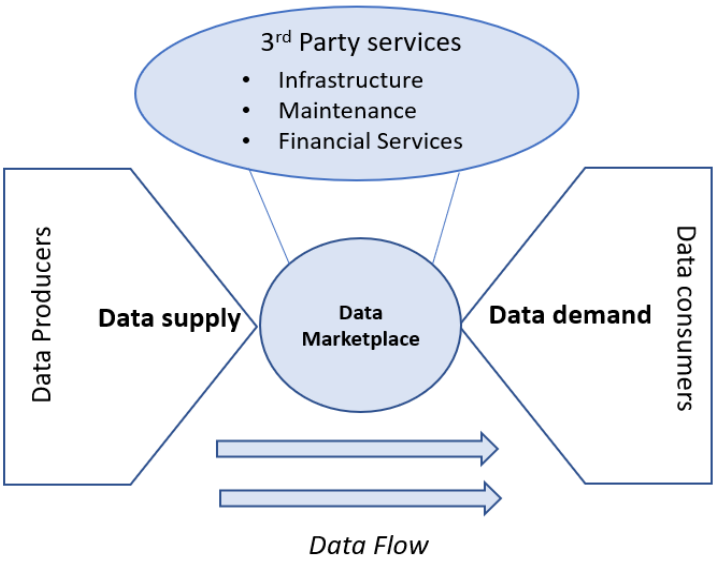

Figure 3: Data flow in a privately-owned data marketplace

(Source: Own analysis)

of companies provide a platform for buyers and sellers to sell their data. They provide infrastructure for the marketplace and maintain it. But they have a major drawback that there is a company acting as a gateway between data producers and data consumers and this company has a commercial interest that differs from other stakeholders despite being bound by a legal agreement [6].

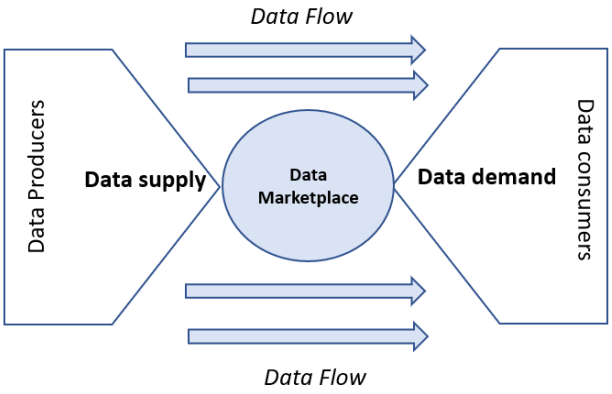

Figure 4 Data flow in a Decentralized Data Marketplace (dDM)

(Source: Own analysis)

Apart from privately-owned data marketplaces there has been also many initiatives for open marketplaces but they are limited in terms of pricing models and also do not provide any incentive for the data producers as the data is mostly free [2].

Thus, a new system that enables direct transactions between the data consumers and data producers is required. A Decentralized Data Marketplace (dDM) is a DM such that, there is no central authority or company which regulates the participants of the market [7]. Figure 4, shows a basic model of a dDM which limits the intervention of third parties and enabling direct transactions between the data producers and data consumers.

1.2 Research Questions and research goal

Thus, in order to increase transparency and direct transactions between data producers and consumers, this thesis aims at designing a decentralized data marketplace (dDM).

Real world example considered for research goals:

The aim of the project Recycling 4.0 is to improve the recycling process by increasing the information flow between the stakeholders. Material requirements are developing dynamically depending on product development and consumer behavior. Accordingly, the recycling system must also behave even more dynamically and predictively and in line with the demands of an upcoming advanced circular economy approach [8].

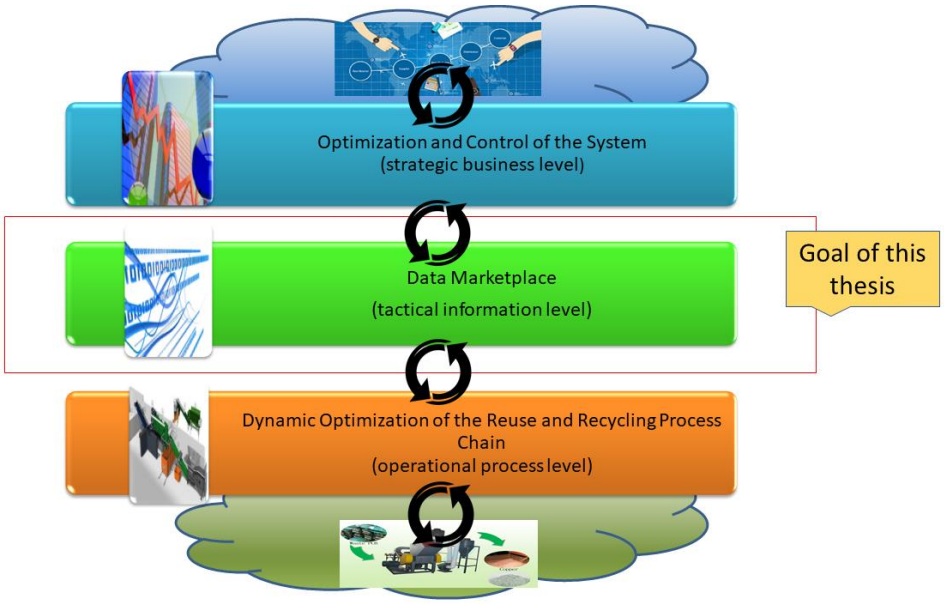

Figure 5 The approach of the project Recycling 4.0 towards an advanced and circular economy [8]

The approach of this project towards an advanced circular economy system consists of three major parts: An open data marketplace to meet information needs, suitable economic assessment and planning methods, and a dynamic optimization of the reuse and recycling process chain, e.g., selection of process steps and their sequence [8]. Fig shows the approach proposed by the project for an advanced circular economy.

This thesis will only focus on the stakeholders (such as Original Equipment manufacturers OEM's, dismantlers etc.) in the recycling industry as sellers and buyers in the data marketplace.

The research goal is divided into the following research questions:

Research Question 1: What are the requirements of a decentralized data marketplace?

To answer this question a literature research with the currently implemented systems must be done. A requirements analysis defining the functional and non-functional requirements is done. Requirements are necessary attributes defined for an system prior to efforts to develop a design for the system. It acts as a transformation between the customer's system need and the design concept. The expected result from this requirement analysis is the definition for creating a system architecture.

Research Question 2: What is the architecture of such a data marketplace?

This research question is focused on designing a system architecture for a decentralized data marketplace. The architecture contains all fundamental specifications and agreements triggered by requirements. In order to design a good system architecture, it is important to choose the right concepts for the defined problem.

Research Question 3: What are the technologies that can be used for development of a decentralized data marketplace?

In order to answer this question various technologies enabling decentralized data marketplace will be evaluated. After suitable technologies are identified, a proof-of-concept implementation based on them, will be implemented in order to evaluate the system architecture.

1.3 Research Method

As described in section 1.3 the focus of this thesis is the stakeholders of the recycling sector. Therefore, the research entry point of this thesis is problem centered.

The thesis follows Design Science Research Model by Peffers, Tuunanen, Rothenberger, and Chatterjee [9] to create and artifact for the mentioned research questions. The resulting artifact is system architecture for a decentralized data marketplace which is evaluated using a prototype. According to the DSRM a prototype is an implementation of an artifact that is demonstrated using a prototype [9]. Scenarios are used to artifact real world situations, so that their applicability can be evaluated.

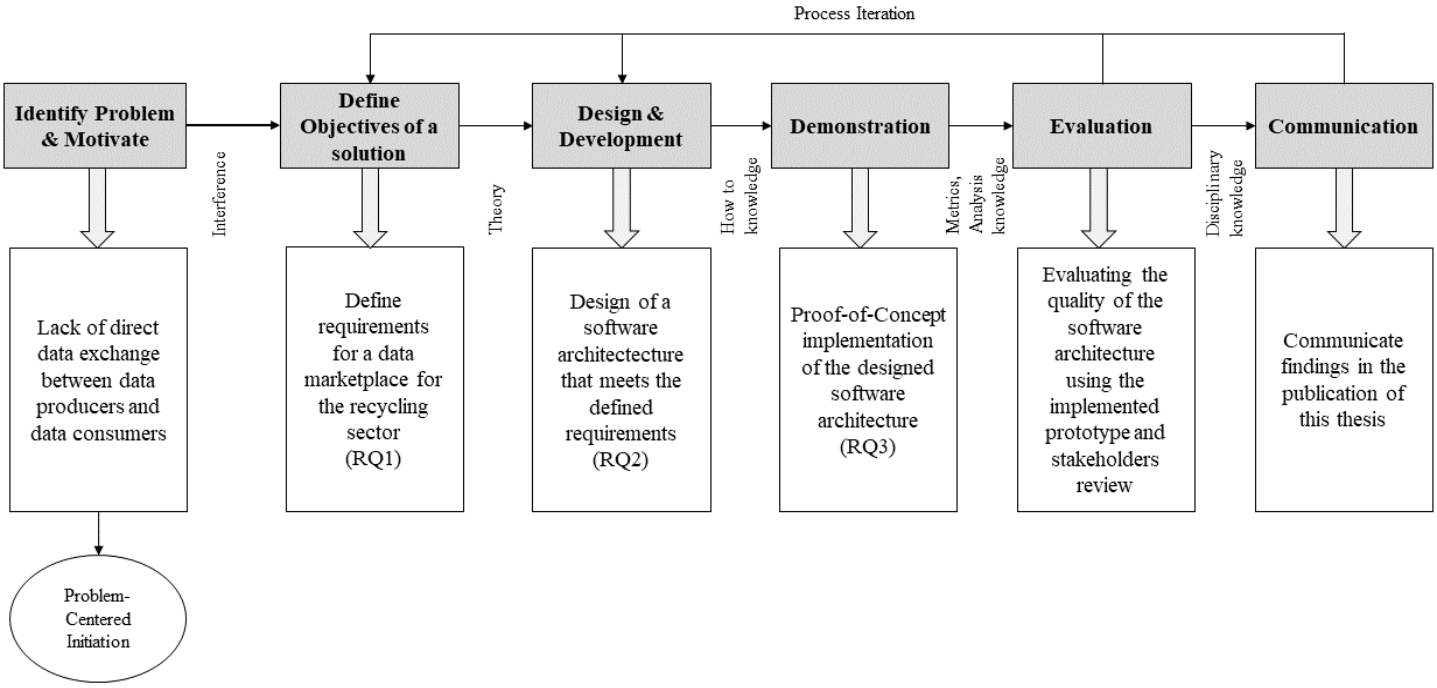

Figure 6 DSRM process decentralized data marketplace

(Source: Based on Peffers et al.[9] *)*

There are six steps in DSRM. The output of each of these steps are used as an input to the other step. Figure 6. shows these steps and the flow of the research process.

Following are the research steps around which DSRM is built:

1. Problem identification and motivation:

The first step in DSRM is to identify and justify the problem**.** Since the problem definition will be used to develop an effective solution, it is important to atomize the problem conceptually so that the complexity of the problem can be captured [10]. Defining a problem helps digesting the complexity of the problem domain. Furthermore, justifying the problem gives insights and reasoning of the research [9]. The motivation of this thesis is described in section 1. Although today more data is generated than ever, and the demand increasing exponentially the current data acquisition methods does not allow direct transactions between data producers and data consumers.

2. Define the objectives for a solution:

The problem motivation and identification are an *inference* to identify the objectives. The objectives can be qualitative or quantitative [11]. Qualitative objectives are like functional requirements describing how the solution can address a problem with its functionalities. Quantitative objectives reflect the non-functional requirements such as security, performance and others. The concrete requirements are described later in chapter 5.

3. Design and development:

The next step is to design and develop an artifact. Using the *theory* created in the previous step the artifact can be created. There are many ways to create this artifact such as a model, an approach, a process etc. This thesis uses the “4+1” view model introduced by Kruchten[12] for design and development of a software architecture for a decentralized data marketplace.

4. Demonstration:

The artifact *(How to knowledge)* and its capabilities can be demonstrated using various methods such as experimentation, simulation, case study etc. [9]. In order to present the quality of the proposed software architecture, a proof-of-concept prototype will be implemented. The evaluation and the design choice reasoning will be explained later in chapter 7.

5. Evaluation:

The performance and applicability of the artifact in solving the problem is evaluated using metrics, analysis and knowledge gained from the demonstration [11].In order to evaluate the artifact presented in this thesis, the requirements presented in chapter 5 will be compared to the designed software architecture. It must be determined whether the artifact was able to solve the requirements. The proof of concept prototype will be tested if the identified requirements are achieved.

1.4 Structure of this thesis

Following the introduction, section 2 describes the approach used in this thesis. Section 3 gives an overview about the background of data marketplaces and blockchain. Section 4 outlines the state of the art and related work. In section 5 the requirements for decentralized data marketplaces are analyzed. Next, in section 6 the proposed system architecture based on the requirements is presented. The design architecture for implementation of the proposed system architecture is described in section 7. A proof-of-concept implementation is also shown in Section 7. Finally, section 8 presents the identified challenges in order to establish a decentralized data marketplace, summarizes the results of this thesis and gives insights in the future work.

2. Approach

In recent years a new technology called blockchain evolved which has the potential to provide a trustworthy, secure platform for peer to peer transactions. Blockchain enables distributed, transparent way for communication. On an abstract level blockchain is a distributed ledger which allows users to send data and verify it without the need of a central entity [13]. Blockchain first caught attention through Bitcoin [7] which enabled users to send money to each other directly without a need of financial institutions. But Bitcoin is just one example of how the underlying technology blockchain can enable peer-to-peer trading. The applications of blockchain expands far more beyond just cryptocurrencies.

In order to enable direct transactions different user, this thesis explores how decentralized data marketplaces can be established. Considering the benefits and possibilities enabled by blockchain in this thesis blockchain technology is taken as an approach for establishment of decentralized data marketplaces.

[7] Bitcoin (₿) is a cryptocurrency, a form of electronic cash. It is a decentralized digital currency without a central bank or single administrator that can be sent from user to user on the peer-to-peer bitcoin network without the need for intermediaries.

3. Background

3.1 Data Marketplaces

While the practice of selling data has existed for a long time, data marketplace is a relatively new business model and refers to mediation service for data-exchange [14]. But in order to understand data marketplaces it is important to understand some common economical terms. In economics the widely accepted definition of a *market* is- a concrete place for interaction between sellers and buyers where the interactions determine the price and quantity of good or service [15]. Thus, this implies that market commonly focuses on one product. In contrast *marketplace* provides infrastructure for trading. Marketplaces are concrete locations that facilitate market [15]. This means that the difference between a market and a marketplace can be attributed to the level of abstraction. A marketplace is the infrastructure that enables the abstract concept of a market [15]. A market serves three main function: First, it serves as an institution i.e. It assigns roles such as buyers and seller. Provides trading protocols and governs behavior of the participants. And finally, a market defines process of transactions [15]. According to Schmid and Lindemann, transactions are divided into three phases: Information phase, agreement phase, and settlement phase [16]. Figure 7. shows these phases.

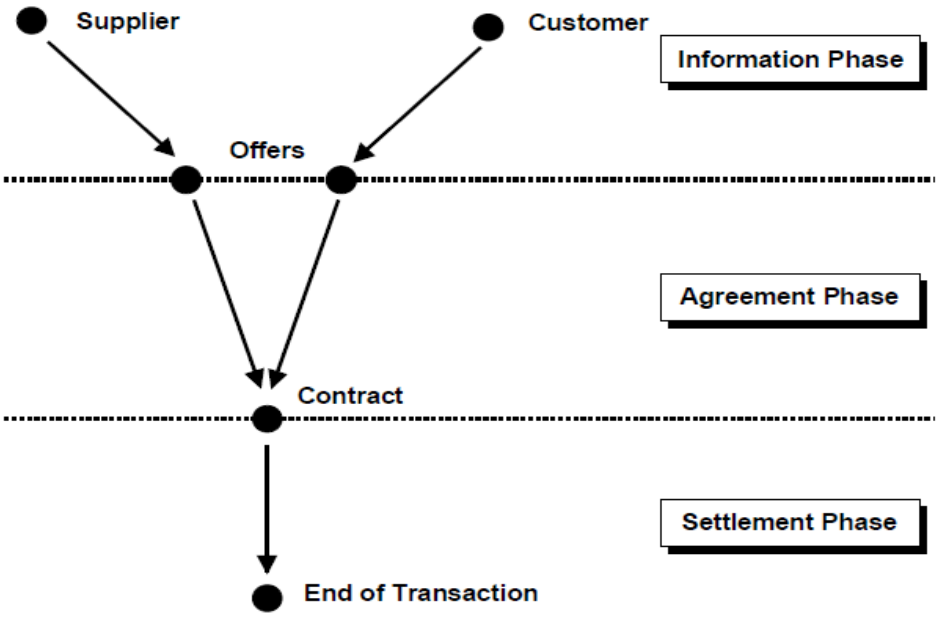

Figure 7 Phases of market transaction

(Source: Schmid and Lindemann [16]*)*

In the *information* phase the suppliers and customers gather overview of the general business environment, technology and other information such as goods and services [16]. After a market participant submits an offer, the information phase ends and the agreement phase starts. In the *agreement* phase, negotiations about the transaction conditions are made. This will allow an agreement to be made [16]. The result is a legal-binding contract representing the agreement between the market partners. In the settlement phase, the agreed-upon terms of the contract are fulfilled. The settlement phase can vary depending upon the type of goods. E.g. for the physical goods it may include packaging, transport etc. [16].

An electronic market as an electronic medium is based on the new digital communication and transaction infrastructure [16]. Following the previously introduced definitions of market and marketplace, an electronic marketplace commonly also known as e-commerce, is an infrastructure or concrete agency that allows participants to carry market transaction via an electronic medium [15]. Just like any other electronic marketplace a data marketplace is a platform which enables convenient buying and selling of a product- in this case data.

There are many definitions of data marketplaces. In [17], data marketplace is defined as a platform which anybody or at least a great number of potentially registered clients can upload and maintain data sets. Access to use of data is regulated by different licensing models. Whereas in [7], a data marketplace is defined as a platform for trading of information which provides- infrastructure for free market i.e. sellers can offer their data in exchange of money, Allows and data item to be evaluated and valued, gives incentives to honest players and provides quality of data. In this thesis, we define data marketplaces as trading platform to sell and buy data.

3.1.1 Types of data marketplaces

There are various models of electronic marketplaces possible based of various factors such as ownership, business models, market strategy etc. The structure of marketplaces affects how people interact on the platform [11]. Nowadays a bunch of various electronic marketplaces – so called electronic commerce platforms – already exist, like for instance eBay, Amazon or Alibaba. These electronic marketplaces are platforms or infrastructure that allows

participants to meet and perform their desired market transactions through an electronic medium. Thus, a data marketplace is also categorized as a form of electronic marketplace. Electronic marketplaces exist in different shapes and can be categorized along various dimensions. As a result of the overlapping definitions of electronic marketplaces, the categorizations are equally confusing. Each model uses different definitions which makes a general classification of the various forms of business models difficult [15]. In [15] a comprehensive model incorporating various dimensions for categorizing electronic marketplaces is proposed. For the categorization of electronic marketplaces in our work we consider this model.

In this model first, providers are placed on a scale of orientation between hierarchy and market [15]. Economies have two basic mechanisms for coordinating the flow of materials or services through adjacent steps in the chain: markets and hierarchies. Markets coordinate the flow through supply and demand forces and external transactions between different individuals and companies. Market forces determine the design, price, quantity, and target delivery schedule for a given product that will serve as an input into another process [15]. Hierarchies, on the other hand, coordinate the flow of materials through adjacent steps by controlling and directing it at a higher level in the managerial hierarchy, rather than by letting market transactions coordinate it. Managerial decisions, determine design, price, quantity, and delivery schedules at which products from one step on the chain. Thus, all transactions between suppliers and buyers can be classified as either hierarchical or market based [15]. Furthermore, marketplaces are categorized in based on their ownership, which can be (a) private, i.e., owned by a single company (seller or buyer); (b) consortia-based, i.e., owned by a small number of companies (seller or buyer); and (c) independent, i.e., the marketplace is run as a platform without any connection to sellers or buyers [15].

Based on these three dimensions market, hierarchy and ownership the model proposed by [15] differentiates six business models. As shown in the Figure 8. at the hierarchy level is the privately-owned platforms.

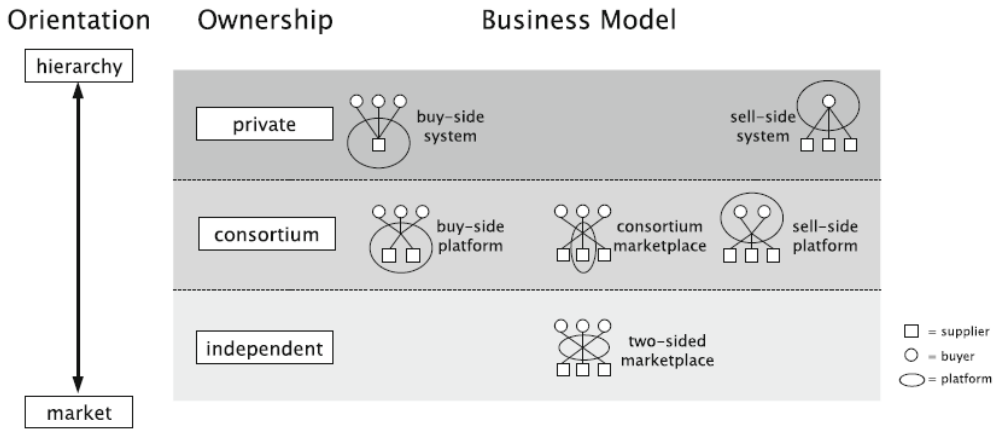

Figure 8 Hierarchy of marketplace structures

(Source: Stahl, Schomm, Vossen, and Vomfell [15]*)*

These types of business model typically facilitate the selling and buying of its owner i.e. a company and only allows one-to- many and many-to-one relations. In between are the consortia- based platforms, these models are typically a collaboration between various companies and facilitate their buying and selling methods. At the market-level many-to-many marketplaces are usually operated by the independent parities and have minimal entry restrictions [15].

3.2 Blockchain

As the approach of this thesis is to design a system architecture for decentralized data marketplace using blockchain, this chapter is intended to give a broader definition of what blockchain technology is, its characteristics and what makes it so special. Although blockchain is rather new technology the core ideas behind this technology emerged in the late 1980's and early 1990's. In 1998 Leslie Lamport published a paper describing the Paxos protocol [13]. This paper describes a consensus model for reaching agreement on a network of a computers, where the computers or the network itself might be unreliable[18] . In 1991, a signed chain of information was used as an electronic ledger for digitally signing documents in a way that could easily show none of the signed documents in the collection had been changed [18]. In 2008 both these concepts were combined and applied to electronic cash and described in the paper *Bitcoin: A Peer to Peer Electronic Cash System* [19], by a pseudonymous author Satoshi

Nakamoto. Based on this paper later in 2009 the first Bitcoin cryptocurrency blockchain network was established. Bitcoin was just the first of many blockchain applications [18]. Although many use blockchain as a synonym it is very important to understand that, bitcoin is a cryptocurrency, or an application enabled by blockchain. Blockchain at its core is a distributed and decentralized open ledger that is cryptographically managed and updated various consensus protocols and agreements among its peers [20].

3.2.1 Distributed systems

In order to understand blockchain it is very important to understand distributed systems as blockchain on its core is a distributed system. More precisely decentralized distributed system. Distributed systems are a computing paradigm whereby two or more nodes work with each other in a coordinated fashion in order to achieve a common outcome and it's modeled in such a way that end users see it as a single logical platform [13].

CAP theorem also known as Brewer's theorem was originally introduced by Eric Brewer and proved by Seth Gilbert and Nancy Lynch. This theorem states that any distributed system cannot have Consistency, Availability, and Partition tolerance simultaneously [21]. CAP theorem defines the properties as follows:

1. **C**onsistency- Every node has the latest copy of data.

2. **A**vailability- Every node is accessible and responses when required.

3. **P**artition tolerance- The system continues to function correctly even if a group of nodes fails.

Blockchain sacrifices consistency in favor or availability and partition. In blockchains Consistency (C) on the blockchain is not achieved simultaneously with Partition tolerance (P) and Availability (A), but it is achieved over time [13]. This is called *eventual consistency,* where consistency is achieved as a result of validation from multiple nodes over time [13].

3.2.2 Elements of Blockchain

Blockchain is a layer of distributed peer-to-peer network running on the internet. The network view of a blockchain can be seen in Figure 9. Each of the components shown in figure 9, are discussed in detail later in this section.

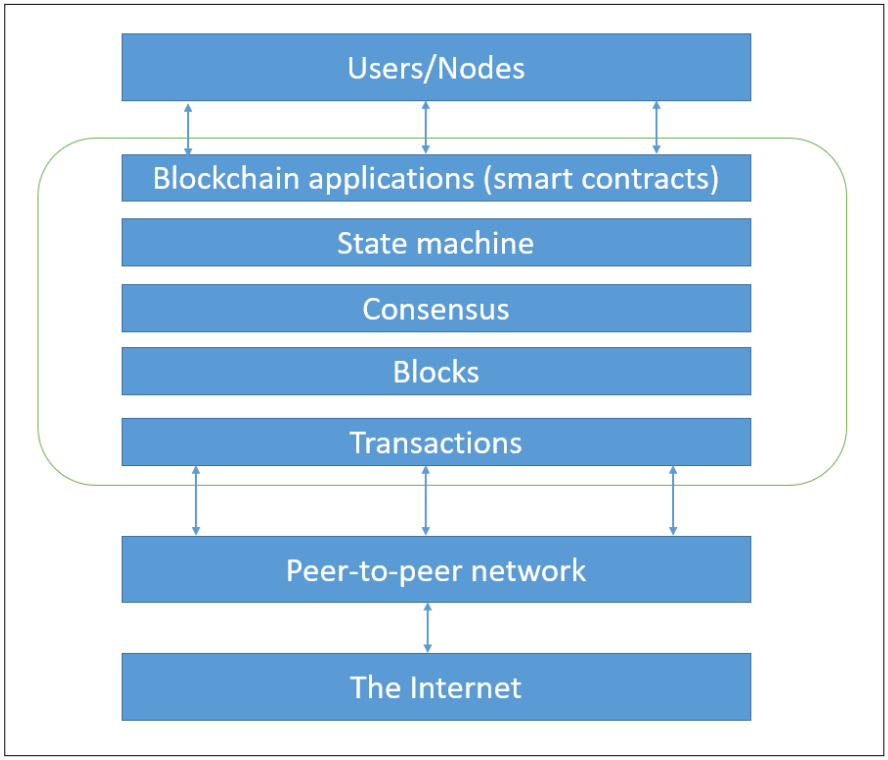

Figure 9 The network view of a blockchain

(Source: Mastering blockchain [13]*)*

Blockchain can be seen as HTTP or TCP/IP running on the internet [13]. In non-technical terms blockchain can be defined as a platform where users can exchange messages or values in form of transaction without a need of a central authority. This makes blockchain a decentralized mechanism which can run without central authorities.

Blockchain technology consist of various elements. Most of the public or open public blockchain consists of the following elements. Some elements many vary in different blockchain applications, but these are the basic elements which most of the blockchains have.

1. A peer-to-peer network

For traditional client-server application, there is a central server and to which systems connect, these systems are known as clients. In client- server application the clients are connected to a central server which acts as a central communication point. Whereas peer-to-peer networks do not distinguish between clients and servers. All the nodes are equally entitled as peers and can communicate with one another without the need of central coordination via servers. Figure 9 shows a pictorial difference between client server and peer-to-peer network. As seen in the figure the systems in client server network communicate via a central server whereas in in peer-to-peer network all nodes can send information to each other directly.

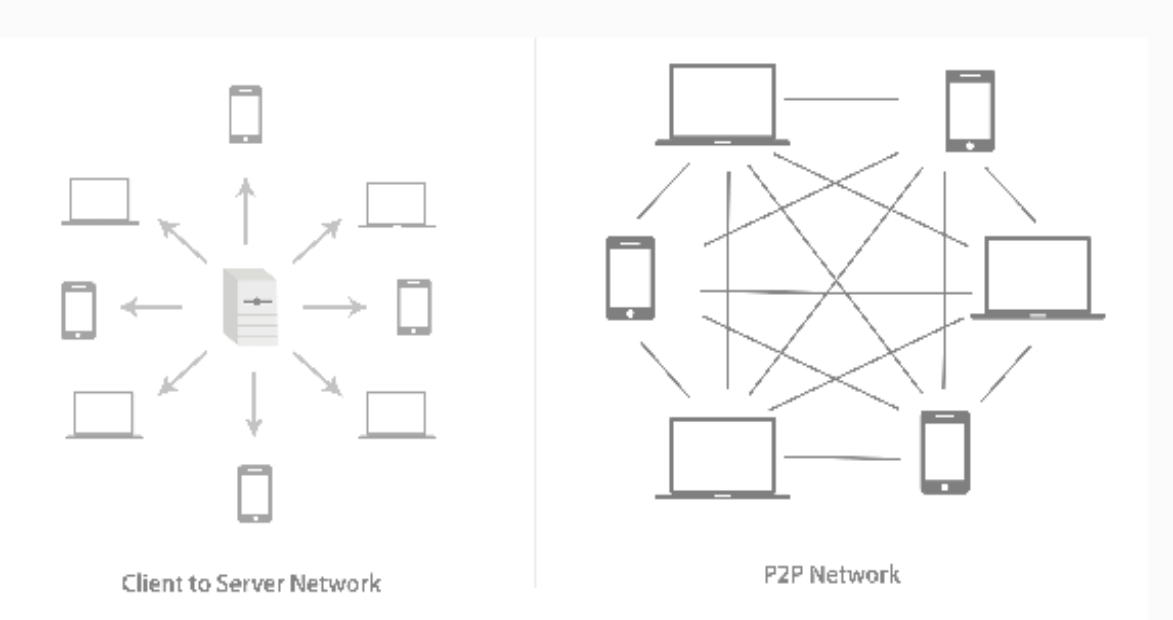

Figure 10 The pictorial difference between a client-to-server network and a peer-to-peer network

(Source: Blockchain Quick Reference [20]*)*

Blockchain is a peer-to-peer network which allows users to be able to share information directly to one another [13]. A distributed network is like peer-to-peer network, but it also allows information to be shared across various user. Blockchain is a peer-to-peer distributed network which allows users to share information directly with on another directly and allows the information to be stored across various nodes [20].

2. Transactions

Messages in blockchain are sent in form of transactions. It is a fundamental unit in blockchain. A transactions represents a transfer of value from one address to another [13].

3. Block

Multiple transactions are encoded in a block. A block is comprised of block header and list of transaction. The first block is known as *genesis block* [13]. The blocks are stored in a Merkle tree formation. Each block includes a hash to previous block forming a chain and link to one another. The block header contains various information such as hash to previous block, Merkle root, timestamp, nonce etc. Figure 11 shows the formation of a block containing the Merkle root and Merkle tree.

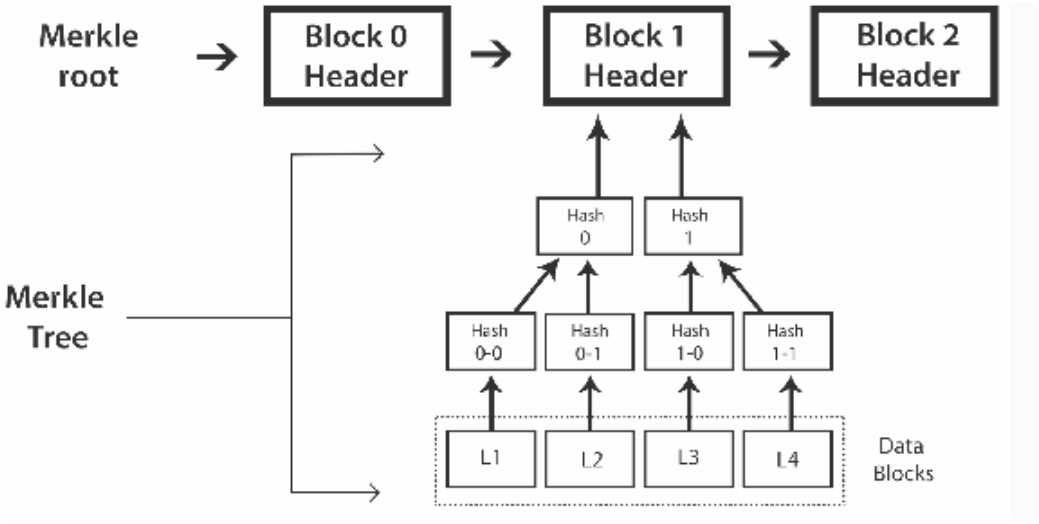

Figure 11 The formation of block headers and what comprises the Merkle root and the Merkle tree

(Source: Blockchain Quick Reference [20] *)*

A Merkle tree is a binary tree in which the inputs are first placed at the leaves i.e. nodes without any children's, and then the values of pairs of child nodes are hashed together to produce a value for the parent node until a single hash value known as Merkle root is achieved [22]. Merkle trees resemble tree like structure, at each step of the tree the root node contains hash data of its children [20] .

4. Addresses

Addresses are unique identifiers required in transactions [23]. An address is used to send data to other addresses. For e.g. in bitcoin addresses are used to send bitcoins from one address to another.

5. Wallets

A wallet is a digital wallet which is used to store the public and private keys along with transaction addresses. In order to send a transaction a receiver's address is required, this address is a public key whereas to finally send the transaction and authenticate it, the sender as to sign it using the private key. This ensures that only authenticated users send a transaction. Along with storing the public key and private key of use, wallets also store a list of carried out transactions. There are many types of wallets available the most popular ones are software, web, paper and hardware wallets. In a software-based wallet, where the private key is stored on the user's machine. Web based wallets are cloud based and can be accessed from anywhere. The private key for a paper wallet is printed on a paper whereas the private key in a hardware wallet is stored on a small portable hardware [20].

6. Consensus mechanism and rules

Consensus is a process by with distributed systems reach an agreement amongst the nodes. Reaching on a common agreement in distributed systems is a challenge as there is no central authority to decide the latest sate. In blockchains consensus is a critical property.

Thus, Blockchain uses consensus algorithms for reaching a common state and still maintain decentralization.

The most widely used consensus mechanisms are the following:

- Proof of Work (PoW)
- Proof of Stake (PoS)
- Delegated Proof of Stake (DPOS)
- Proof of activity (PoA)

Along with consensus mechanisms a set of consensus rules needs to be defined too. These rules governs what's valid as a transaction and what makes a valid state transition [23]. which

can vary throughout different blockchain applications. Proof-of-work consensus mechanism used in bitcoin and many other cryptocurrencies relies on proof that enough computational resources have been spent before proposing a value for acceptance by the network [13].

7. Chain of cryptographically secured blocks

The transactions encoded in blocks and verified by all the nodes thus reaching consensus. The block doesn't gets added to the chain if the nodes did not accept it. The rules for reaching consensus are pre-defined by the implementation. This chain acts as journal for all the accepted and verified transactions. Figure 12 shows an example of a cryptographically secure chain of blocks. Each accepted block is added to the chain. Every block contains a hash of the previous block. Given an input a fixed length of hash can be generated. Various hash functions such as SHA1, SHA-2, SH-3 MD etc. are available. One important property about hash functions is that once the input is changed the generated output also changes.

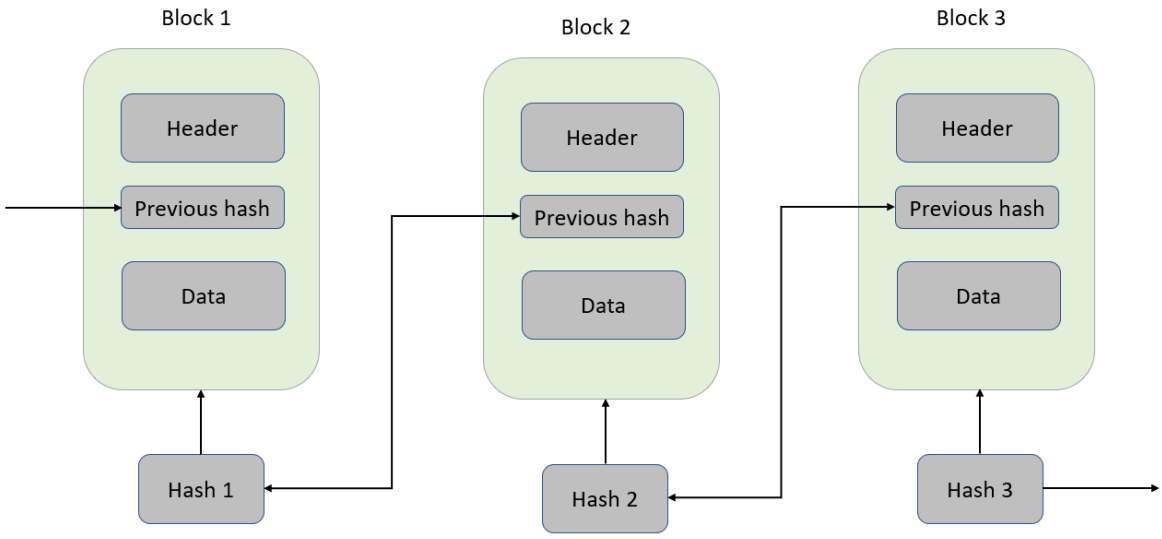

Figure 12 Example of a cryptographically secured chain of blocks

As the chain of blocks includes the hash of previous block too. Any attempt to change any block would invalidate the whole blockchain thus making it cryptographically secure chain.

8. State Machine

A state machine processes transaction according to the defined consensus rules [13]. A blockchain can be viewed as a state transition mechanism, where the state of transactions change from its initial state until the validation by nodes and transaction execution.

9. Nodes

There are many roles a node can perform. The roles of the nodes can also vary according to the consensus rules and the blockchain implementation. A node can just connect to the network to perform transaction. It can also save the full copy of the blockchain to facilitate consensus and validating block, these nodes are known as full nodes. On the other hand, a node can also be a light node, connecting to the full nodes to get the current state of the blockchain. Nodes play a very role in blockchains to secure the chain.

3.3.3 Types of Blockchains

From its first big implementation as a use of peer-to-peer cash, blockchain evolved in many different types. These types also enabled blockchains to perform many more things beyond cryptocurrencies. Blockchains are mainly classified into four types- Public, Private, semi-Private and Consortium [20]. Depending on the application requirement the type of blockchain can be selected. Figure show the types of blockchain networks currently available and proposed.

1. **Public Blockchains:** In a public blockchain anyone in the world can be part of the network. Anyone can be a part of the transaction process as well as be a part of the validation and consensus mechanism. It is a completely open ledger and is also sometimes called permission less ledgers. One popular example of a public blockchain is bitcoin.
2. **Private Blockchains:** Unlike public blockchains private blockchains are not open to everyone and are limited to a certain group of people or one organization. E.g. Corda

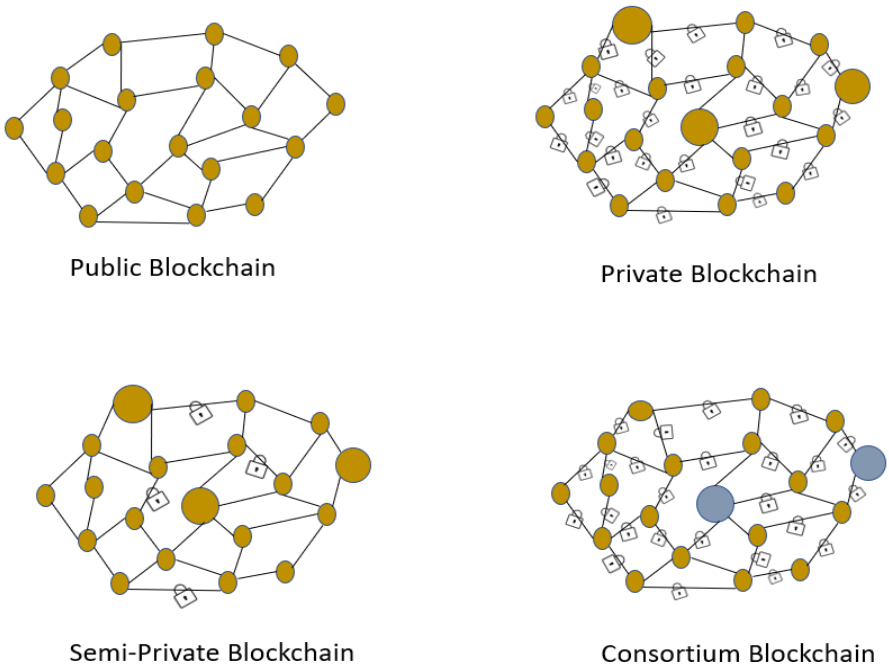

Figure 13 Types of blockchain networks

(Source: based on Blockchain quick reference [20] *)*

3. **Semi-private Blockchain:** Like private blockchains semi-private blockchains are also run by a single organization or a group of individuals. But unlike private blockchains, semi-private blockchains has a public part with is open for anyone for participation.
4. **Consortium Blockchain:** The consensus power i.e. validation of blocks in consortium blockchains is restricted to just some pre-selected nodes, the other nodes only can participate in transactions. This type of blockchains are also known as permissioned private blockchain.

3.3.4 Tiers of blockchain technology

Although many concepts and elements of blockchain have been around for a while, blockchain itself is a rather new and evolving technology. After its first successful implementation in the form of bitcoin, the potential of blockchain was realized beyond cryptocurrencies enabling many other applications. Some have already been realized while some can be envisioned for the future based on the current rate of advancement in the blockchain technology [13]. Melanie swan in her book *Blockchain, Blueprint for a New*

Economy categorizes the blockchain in three tiers- Blockchain 1.0, Blockchain 2.0, Blockchain 3.0,based on its capabilities and applications [24]. In addition to this Imran Bashar also describes one more tier- generation X as the future of blockchain when the technology evolves to be advanced enough [13].

1. **Blockchain 1.0:** The generation 1 of blockchain is *currency*, the deployment of cryptocurrencies in applications related to cash [24]. Bitcoin was the first implementation and all the other alternative coins and cryptocurrencies can be classified as generation 1 of blockchain.
2. **Blockchain 2.0:** The second generation is *contracts* and use of blockchain for economic, market and financial applications [24]. In this tier contracts are introduced for applications such as bonds, loans, smart property, smart contracts etc.
3. **Blockchain 3.0:** Blockchain 3.0 extends to *applications* which are beyond just financial services to more generalized applications such as health, government, science, art etc. [24].
4. **Generation X (Blockchain X):** Although in the past few years the technology has evolved a lot, many visions and propose the future as blockchain singularity. This means the use of blockchain service available to anyone and can be used easily just like google search engine. A public open general purpose ledger agents running on blockchain capable of making decisions and interacting with other intelligent agents of behalf of humans is predicted [23].

3.5 Smart contracts

The concept of smart contracts was first conceived by researcher Nick Szabo in the mid-1990s. Nick Szabo describes smart contracts as set of promises, specified in a digital form including protocols within which parties will perform what promises [25].Thus, the most important components of a smart contract are:

- A set of rules or promises
- It is in a digital form
- The Protocols for communication and performance are defined
- Performance of actions is triggered automatically.

Blockchain provides platform to run smart contracts. Thus, enabling automatic execution of a contract on behalf of users. But it is important to note that smart contract and blockchain are different ideas. A blockchain can exist without smart contracts too e.g. bitcoin and a smart contract can exit without blockchain too. However, smart contracts and blockchain together enable many new possibilities which were not being achieved until now [13]. Blockchain provides two out of the four important components for smart contracts I.e. a protocol for communication and performance of actions between various parties and a digital form [20].

As already mentioned, in blockchain a user is identified using an address. Like user addresses smart contracts on blockchain also get an address through which other smart contracts and users or applications can communicate with them.

In terms of Ethereum [8] a smart contract is an immutable computer program that runs deterministically in context of a Ethereum virtual machine which is a part of the Ethereum network protocol [23]. Thus, in context of Ethereum smart contracts are computer programs which one deployed cannot be modified unlike other software programs. The only way to modify a smart contract is by invoking a new instance [23]. Thus, smart contracts on Ethereum should be well tested before deployment. The outcome of the execution of the smart contract is the same for everyone given the state of the blockchain at the time of execution, this makes it deterministic. The most popular language used to write smart contracts on Ethereum is solidity.

3.6 On chain and off chain storage

Blockchain is a combination of various disciplines and computing concepts such ss cryptography, economics, decentralized storage, decentralized computing, peer-to-peer networking etc. This combination has positioned blockchain as new technology. Blockchain has introduced many unique properties such immutability and transparency. It achieves decentralization by storing data in blockchain. Every full node store all the data of the blockchain. But this by design makes blockchains not suitable for large amount of data. As the data in the blockchain increases more and more storage by the nodes is required.

[8] Ethereum is a global, open-source platform for decentralized applications. On Ethereum, you can write code that controls digital value, runs exactly as programmed, and is accessible anywhere in the world.

Thus, in the past years many initiatives and research has been carried out to identify what should be stored in the blockchain and what not.

On-chain data – in the form of confirmed transactions organized in ordered blocks [26]. Transaction validation, consensus protocols, and decentralized program execution may, however, describe a communication and execution overhead [26]. This makes blockchains less scalable. The objective for off-chaining data and computation is to reduce or to overcome such limitations. By moving data and computation elsewhere off the blockchain, for example, to another datastore, server, or third party [26]. As the data marketplace needs large storage capacity for storing and selling data, not all the data can be stored on-chain in the blockchain. Thus, only some transaction data such as data description, price etc. can be stored on-chain whereas as its more feasible to store the real dataset off-chain. The data marketplace then can be used as a medium for payments and getting access to the off-chain dataset.

Distributed hash tables (DHTs) are one alternative to not store all the data on blockchain. Inter Planetary File System (IPFS) vision's is to provide a decentralized World Wide Web by replacing the HTTP protocol. IPFS[9] uses Kademlia DHT and Merkle DAG (Directed Acyclic Graph) to provide the storage and searching functionality [13]. BigChainDB [10]is another storage layer decentralization project aimed at providing a scalable, fast, and linearly scalable decentralized database . Other platforms such as Storj[11] aims on proving decentralized cloud storage [27].

[9] https://ipfs.io/

[10] https://www.bigchaindb.com/

[11] https://storj.io

4. State of the Art

Real-time data will increasingly turn into a commodity in the coming years. With the aim of providing real-time data, Streamr[12] is a data marketplace market for real-time data [28]. In the data market, anyone can publish events to data streams, and anyone can subscribe to streams and use the data in decentralized apps. Much of the data is free, but where that's not the case, the terms of use are stored in Ethereum smart contracts [28]. In the recent years there has also been many ideas proposed with the aim of giving back the power to the consumers to decide whether they want to share their personal data. One such project is Datum[13]; the Datum Client empowers users to take control of all their data and optionally share or sell their data through the Datum network [29]. The IOTA[14] Marketplace is a decentralized data marketplace that aims to make IOT data available to any compensating party. The Mobility Data Marketplace (MDM)[15]enables different parties to offer mobility data, such as petrol prices, or construction sites on motorways.

[12] https://www.streamr.com/

[13] https://datum.org/

[14] https://data.iota.org/

[15] https://www.mdm-portal.de/

5. Requirement analysis

This chapter focuses on the first research question of the thesis i.e. What are the requirements of a decentralized data marketplace? As this thesis will only focus on the stakeholders of the recycling industry. Requirements are necessary attributes defined for an system prior to efforts to develop a design for the system [30]. It acts as a transformation between the customer's system need and the design concept [30]. Functional requirements describe what the system or software must do [31]. A function is a useful capability provided by one or more components of a system. Nonfunctional requirements specify system properties, such as reliability and safety [31].

At most general level there are three main stakeholders in a data marketplace: **Data Producers** or data providers (*sellers*), **data consumers** (*buyers*) and **data marketplace owners**, those who own and maintain the data marketplace. As in this thesis we explore the architecture of decentralized marketplace, there is no central entity or owner of the data marketplace, the two main stakeholders considered are the sellers and the buyers.

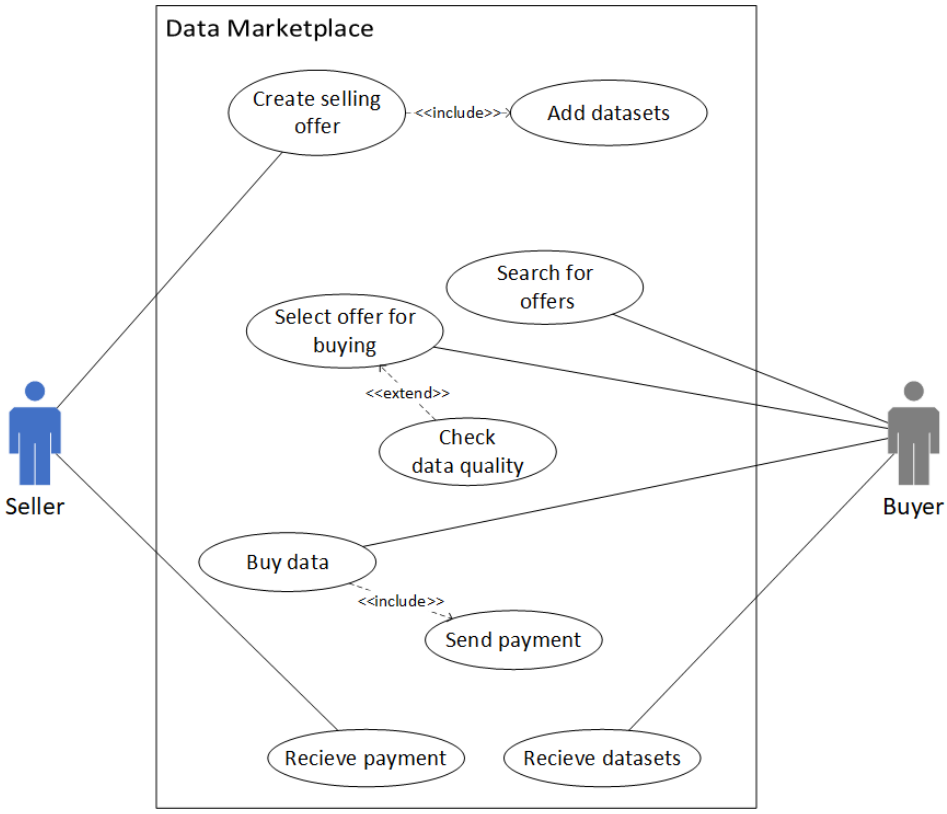

Figure 14 Use case diagram for requirement analysis

(Source: Own analysis)

5.1 Functional requirements

Figure 14. shows the use case for the functional requirements of a data marketplace. Following are the requirements in details.

- **Create selling offer:** The buyer should be able to create a selling offer of the dataset on the data marketplace.

- **Search for offers:** The buyers should be able to search and browse all the active and available selling offers as well as specific data offers.

- **Select offer for buying:** Once a buyer finds a relevant dataset, he/she should be able to read the data description in detail and select it for buying.

- **Buy data:** Once selected, the data marketplace should process the buying order for the buyer and notify the seller.

Send payment: After the buying order is received the data marketplace must validate whether the buyer has enough funds to buy the datasets. If the buyer has enough funds, the data marketplace should process the payment process in the right manner. It is also important that the payment take place in the right way, such as only the amount stated in the offer is deducted from the buyers account.

Receive payments: Once the payment is successfully carried out, the seller should receive the right amount in his/her account. It also must be made sure that the seller who posted the data offer is the one getting the right amount for the corresponding offer.

- **Receive datasets:** when the payment is carried out successfully the buyer should receive the datasets. It also must be validated that the buyer gets the dataset he/she selected for buying.

5.2 Non-functional requirements

- **Availability:** The system should be available and functional for a time period more than agreed operational and performance time also known as uptime.

- **Reliability:** The system should work and perform the intended functions for a specific time interval under the stated conditions.

- **Security:** The system should be protected against malicious attacks.

- **Usability:** The system should be easy to use and easily adaptable by the users.

- **Privacy:** The system should maintain the acceptable level of privacy and protecting the user's information.

Table 1 summarizes of the most important identified functional and non-functional requirements from the data marketplace.

Table 1 Summary of requirements

#	**Name**
1.	Create selling offer
2.	Search for offers
3.	Select offer for buying
4.	Buy data
5.	Send payment
6	Receive payments
7.	Receive datasets
8.	Availability
9.	Reliability
10.	Security
11.	Usability

6. System Architecture

In this thesis the data marketplace is considered as a web application, this explores system architecture of a web based decentralized data marketplace.

This depicts the third parts of the DSRM process. In order to capture a gist of the whole architecture a single view is not enough. Thus, in this thesis the "4+1" view model from kruchten is used to show the system architecture [12]. The model composed of five views. Figure.15 shows the "4+1" view model.

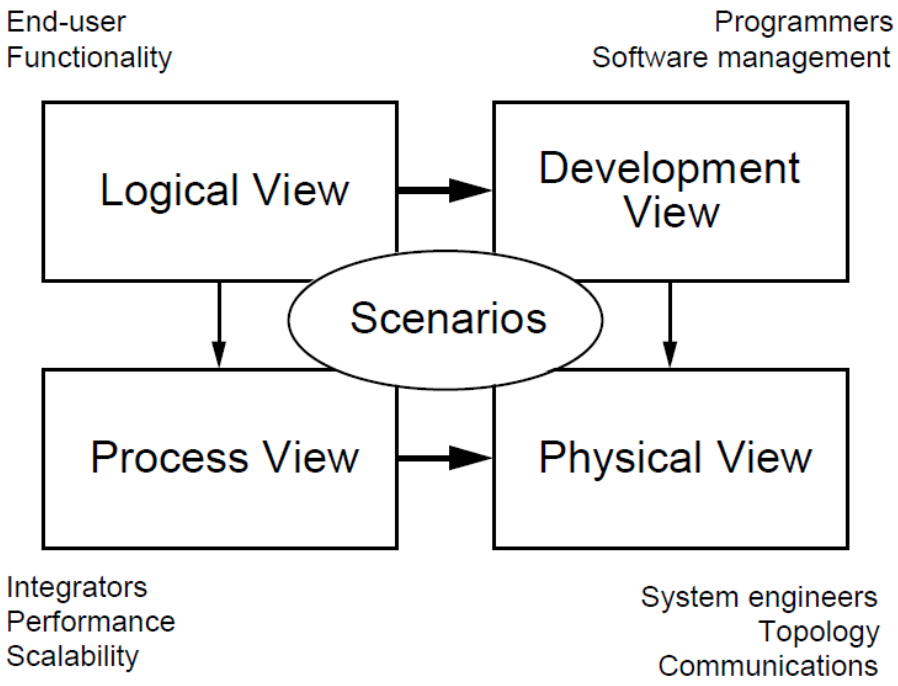

Figure 15 The "4+1" view model

(Source: Architectural Blueprints—The "4+1" View from Philippe Kruchten [12]*)*

The four views are:

1. The Logical View: The logical view primarily focuses on the object model of the design. It also focuses mainly on the functional requirements of the system [12].

2. The Process view: The process view demonstrates the process flow of the system and captures concurrency and synchronization aspects of the design [12].

3. The Physical view: The physical view mostly focuses on the non-functional requirements of the system such as availability, scalability, reliability etc. It maps different parts of the software architecture to the physical hardware the are running on [12].

4. The Development view: The development view focuses on how the software components and modules can be organized. It describes the static organization of the software in its development environment [12].

The description of an architecture—the decisions made—can be organized around these four views, and then illustrated by a few selected *use cases*, or *scenarios* which become a **fifth view** [12].

6.1 Logical View

Figure 16 shows a domain model for the data marketplace.

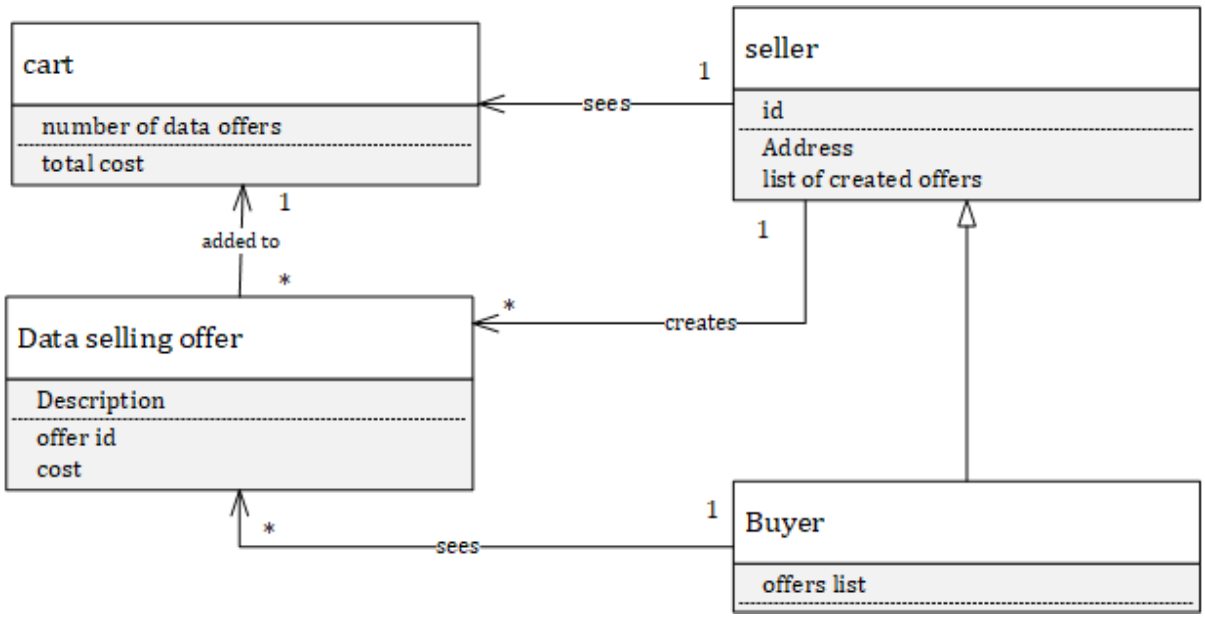

Figure 16 Domain model for data marketplace

(Source: own analysis)

In order to sell data on the data marketplace the seller can create an account. To create a seller account, they choose a name and account address. After that they can create a data offer by giving the data description and price. When the information is added, the offer can be created by a seller. Buyers can buy data by searching for the data offers. Offers consist of the data description and price added by the seller. Once the buyer decides to buy the data the data offer is added in the cart from which it can be bought.

6.2 Process View

The process view shows the typical flow of various processes. The process is a grouping of tasks that form an executable unit [12]. In this section the processes of two main requirements i.e. buying and selling data is shown. In order to show the processes a simplified activity diagram based on Unified Modelling Language (UML) is used. On an abstract level the processes take place between the users and data marketplace.

1. Add selling offer

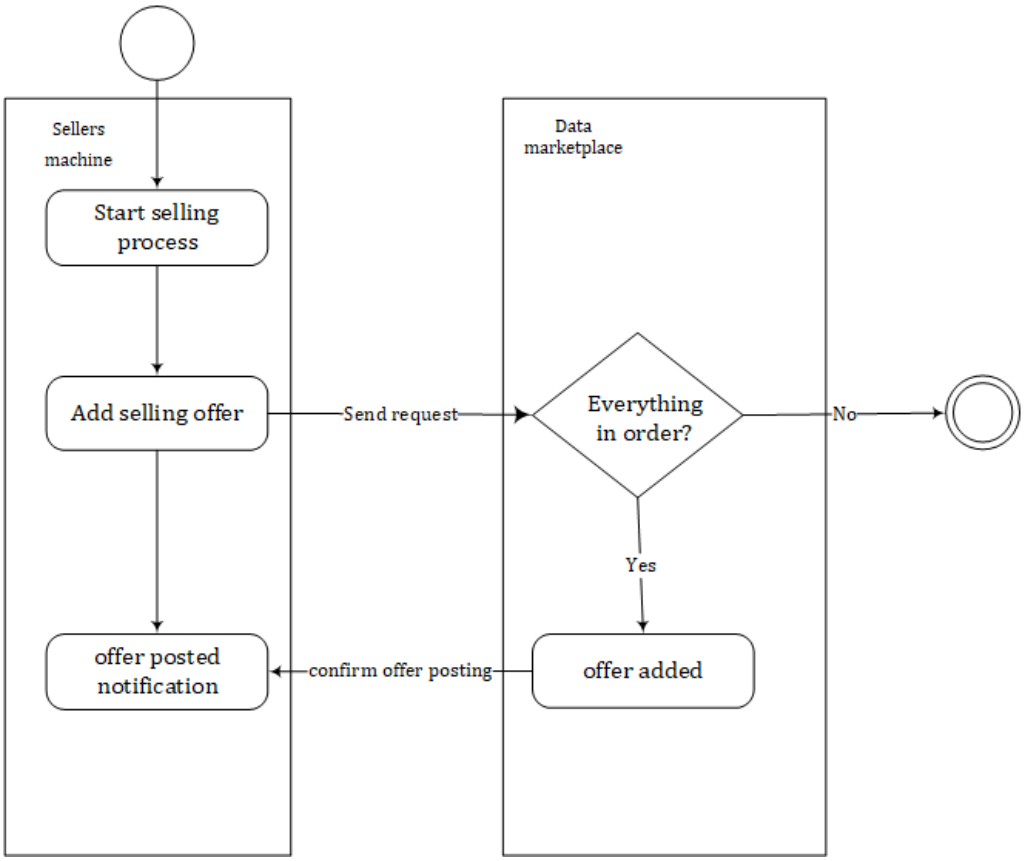

Figure 18 Activity diagram for the process of creating a selling order

(Source: own analysis)

Figure 17. shows and activity diagram for the process of creating a selling offer on the data marketplace. In order to sell data, the seller needs to create a selling offer including all the required parameters such as description, price etc. when the data posting request is created by the seller the data marketplace checks if everything is alright. In case of connection or other failure the request might be cancelled. If everything is alright, the posting offer is accepted, and the selling offer is created.

2. Buying process

The buying process includes the data selection, transaction validation and data transfer. Figure 18, shows the process of buying. In order to buy the datasets, the sellers need to first initiate a buying request. The buying request includes the selected data offer and quantity. If a buying request is initiated, the request is sent to the data marketplace. The data marketplace checks whether the buyer has enough funds for buying the order. If there are insufficient funds in the buyers account, the buyer is notified, and the transaction is cancelled. If the buyer has enough funds and the offer is still active, then further processes are carried out. The required amount for the order is subtracted from the buyers account, the seller is sent the money for the datasets and the buyer receives the datasets. Before the final transaction is carried out and the buyer and sellers are notified about the purchase, it must be validated by the data marketplace that all the information and processes for the purchase as required.

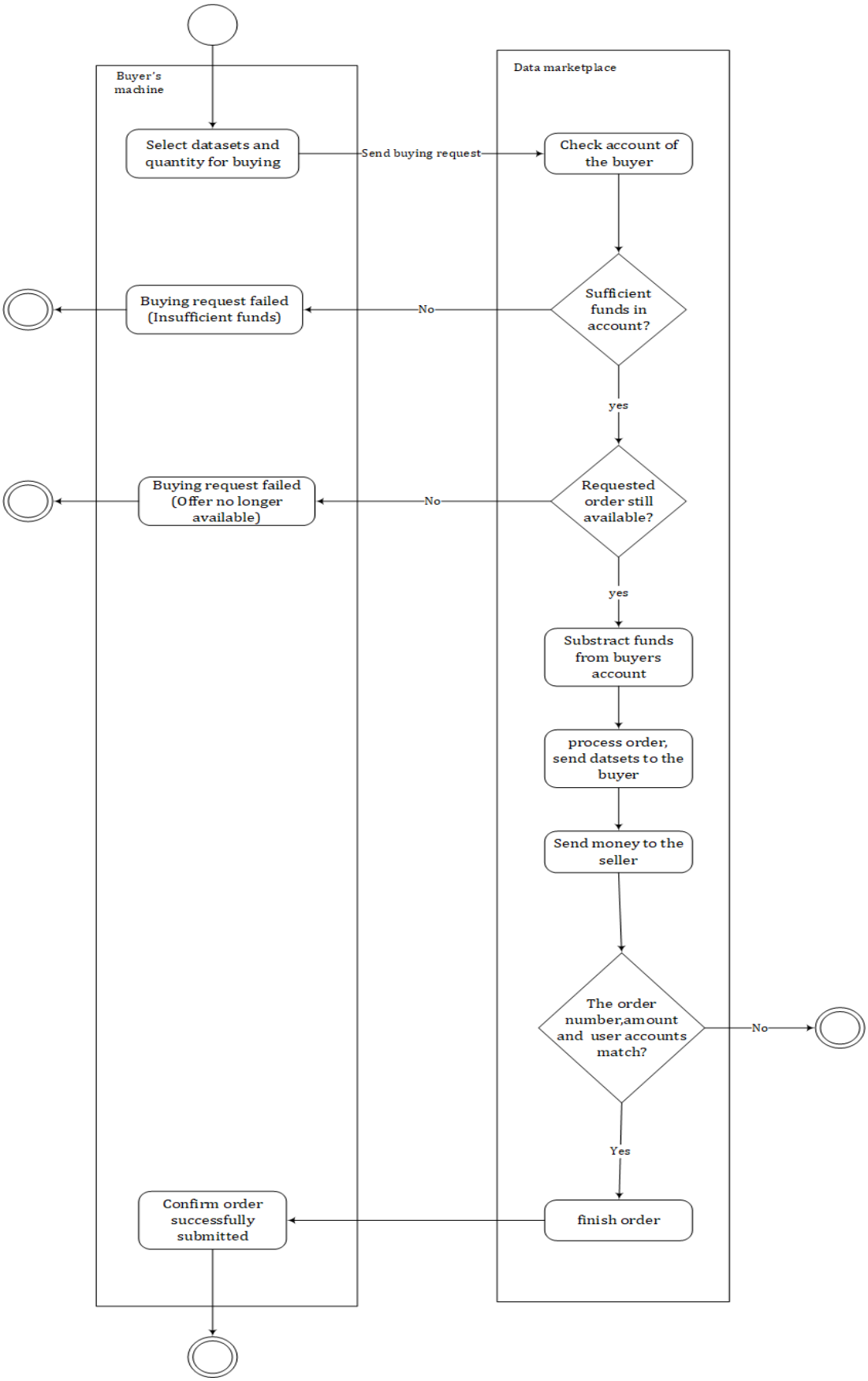

Figure 19 Activity diagram for buying process

(Source: own analysis)

6.3 Development View

The development shows the logical view described in Section 6.1 from the developer's perspective. The whole architecture of the data marketplace is divided into various modules. Each module captures the functional requirements identified in Section 5.1.

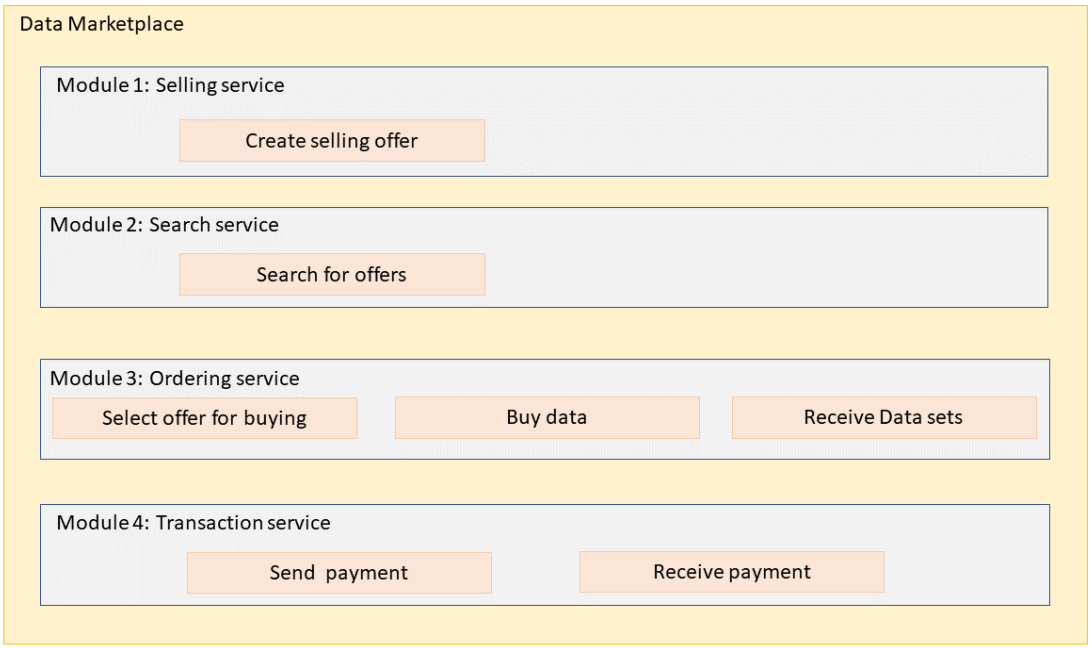

Figure 20 Structure of modules

(Source: own analysis)

The modules follow principle of modularity i.e. the functional responsibilities of the modules should be distinguishable [10]. Separating the functionalities in various modules decreases the complexity and makes it easier to add new functionalities later. Figure. 17 shows the module structure for the data marketplace and shows the functionalities of each of them with respect to the identified functional requirements. The modules are named as selling services, search service, ordering service and transaction service. The selling service includes the functionality of letting the user create a selling offer on the data marketplace. The search service lets the buyer search for active selling offers. With the help of the ordering service the buyer can select and buy data. It also makes surest that the buyer gets the selected data after payment. The transaction service is one of the most important service for an electronic marketplace. The transaction service validates whether the buyer has enough funds to buy data. It also makes sure that the respective seller gets the amount when the buyer sends the

payment. In traditional electronic marketplaces the transaction services are provided by external financial service and institutions like banks, PayPal [16] etc. but in a decentralized blockchain based data marketplace proposed in this thesis the transactions can also be carried out and validated by anyone who is a part of the network. In a blockchain based application the payments can be done by cryptocurrencies such as blockchain, ether etc. It should be noted that the first successful implementation of blockchain was as a new form of peer-to-peer electronic cash system widely known as cryptocurrencies. Thus, blockchain based application do not need external financial entities for the transaction services.

6.3 Physical View

The physical view describes the system architecture from its deployment view and how the physical layers communicate. The software architecture runs on a network of computers or processing nodes, the mapping of these elements is shown in this section. The physical view defines the physical connection between different components.

In this thesis the data marketplace is considered as a web application in order to understand the physical connection of a decentralized web application it's important to understand how traditional client-server-based web application works. Figure 20. shows the architecture of a client-server web application and a blockchain based decentralized application.

[16] PayPal Holdings, Inc. is an American company operating a worldwide online payments system that supports online money transfers and serves as an electronic alternative to traditional paper methods like checks and money orders

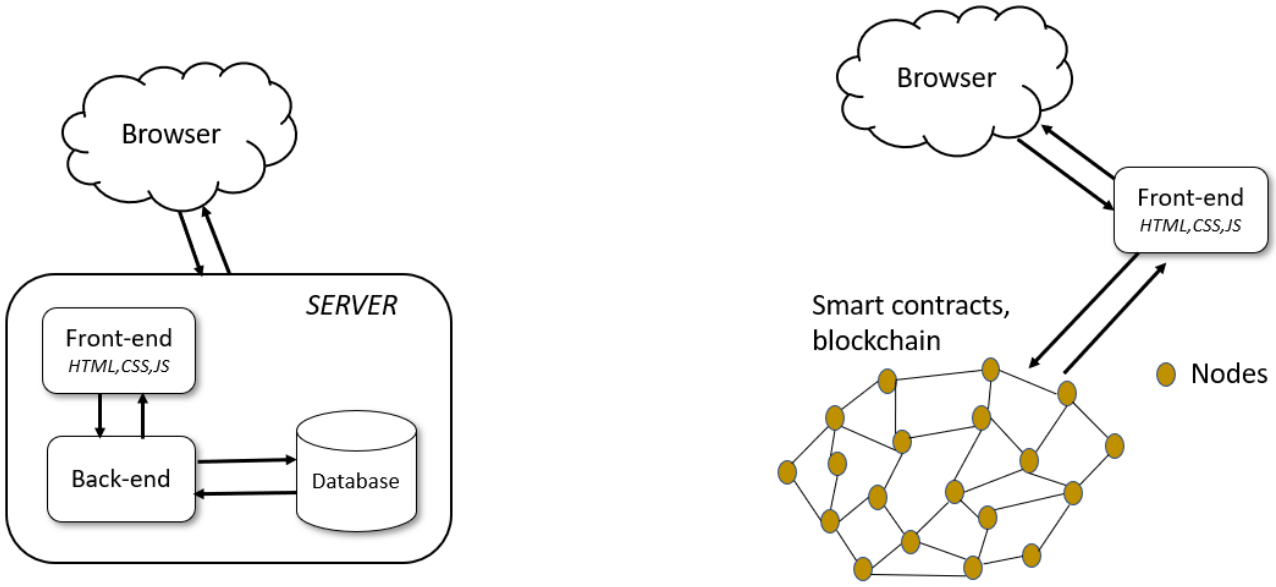

Figure 21 Client-server web application architecture vs blockchain- based web application architecture

(Source: own analysis)

In traditional web applications the user communicates with the system with via a web browser. The client-side files in HTML, CSS and JavaScript which makes the front-end, the backend code responsible for the applications business logic and the database that stores all the information is on the server. This server is a usually centralized entity that full control over every aspect of the application. A blockchain-based application works quite differently. All the code and the data does not lie in centralized server, but it is distributed across the blockchain. The user connects to the application via a web browser but instead of communicating to a back-end web server, the client-side application communicates directly to the blockchain where. The smart contracts are also deployed on the blockchain.

As shown in Figure 21, the architecture of the data marketplace is deployed on three different physical layers- a front-end which runs on the clients browser, the business logic which runs in a state machine of a blockchain specified as smart contracts and an off-chain storage running either on a centralized server or a node of a decentralized storage system. Section 3.6 describes the requirement of an off-chain data storage in blockchain based applications.

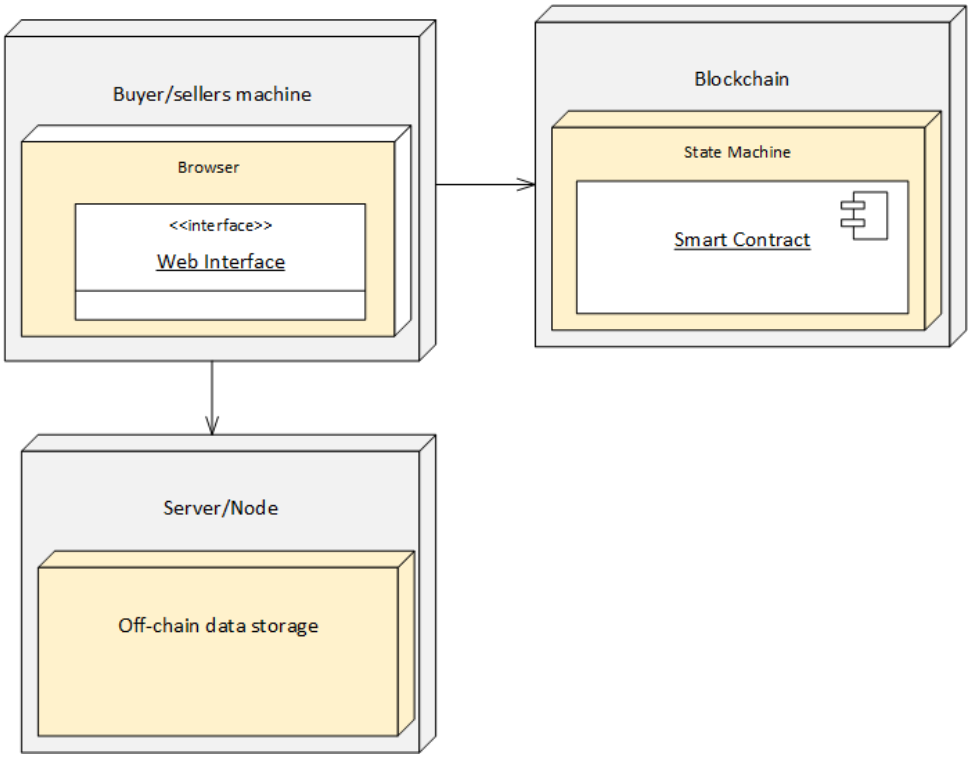

Figure 22 Deployment diagram

(Source: own analysis)

The core processing logic of the application is implemented by smart contracts deployed on blockchain. The smart contracts are executed, and pre-defined events are triggered when the user interacts with the application via the web browser.

7. Implementation

Based on the system architecture and requirements of a decentralized data marketplace, various technologies are evaluated, and a design architecture is proposed in this section. Moreover, to evaluate and demonstrate the system architecture a proof-of-concept is developed. The proof-of-concept uses smart contract to verify the buying process shown in Section 6.2.

Figure 22, shows the design architecture for the decentralized data marketplace.

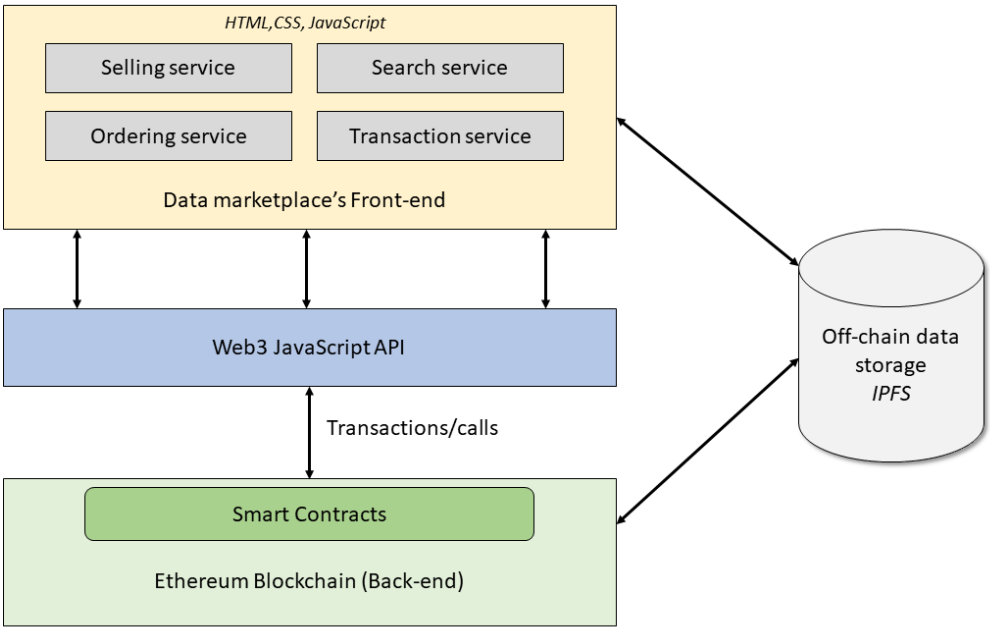

Figure 23 Design architecture

(Source: own analysis)

7.1 Frontend

As the decentralized data marketplace in this thesis is a browser-based web application the front-end technologies selected are HTML, CSS and JavaScript. The front-end plays a very important role in the design architecture as the user interacts with the system through it. Thus, the front-end should be easy to use and adaptable. It should also support the various functionalities required by the users. Thus, as shown in Figure 22, the services such as ordering service , selling service etc. described in section 6.3 are also a part of the front- end.

7.2 Web3

Web3 is proposed as the third version of the web which lets developers create “value” web applications. The third version of web first proposed by Dr. Gavin Wood which represents which focuses on decentralized applications built on decentralized protocols [23].

As shown in section 6.3 blockchain based application architecture is quite different the traditional client-server application architecture. Currently, not all the browsers are compatible to blockchain connections, thus a layer for blockchain connects is required. Currently, web3 is referred as this layer that lets applications interact with the Ethereum blockchain

Web3 is a JavaScript library that can be used to communicate with an Ethereum node via RPC communication [13]. Web3 works by exposing methods that have been enabled over RPC. This allows the development of user interfaces that make use of the web3 library in order to interact with the contracts deployed over the blockchain [13]. There are different aspects of developing blockchain application, one is developing smart contracts that are deployed on blockchain and second is development of web applications or clients to interact with the blockchain. Web3 enables web-applications and clients to interact with the blockchain. The front-end can be connected using the Web3 API to the blockchain.

7.3 Off-chain storage

Blockchains by design are not good for storing large amount of data. As data marketplaces needs to store large datasets for selling its not feasible to store the data on-chain. Section 3.6 presents more details about on-chain and off-chain data storage.

Although there are many possibilities to store data off-chain using centralized and decentralized methods for the design architecture Interplanetary File System (IPFS) is selected.

IPFS is designed to be a decentralized, peer-to-peer file system which uses Directed Acyclic Graph, or DAG [13]. With IPFS, decentralized applications have a way of storing and accessing this content without losing decentralization.

When the data is stored on IPFS, it returns a hash. The data can be retrieved only using this hash. Thus, the data marketplace can be used as a platform to exchange the hashes. This reduces the on-chain storage on the blockchain.

7.4 Backend

Ethereum blockchain is currently the largest community based-blockchain project backed by bitcoin [13]. Bitcoin was the first biggest blockchain application for peer-to-peer electronic cash system but unlike bitcoin Ethereum is a blockchain-based system with special scripting functionality that allows other developers to build decentralized and distributed applications on top of it. Thus, Ethereum has been selected has a blockchain for the proposed design architecture shown in Figure 22.

Components of Ethereum

1. **Ether:**

   Ether is the native cryptocurrency to the Ethereum blockchain.

2. **Ethereum Accounts:**

   Ethereum accounts play a very important role in Ethereum blockchain. In Ethereum there are two types of accounts: *Externally owned accounts* (EOA) and *contract accounts*. An EOA is controlled by private keys whereas the contract accounts by their respective contracts. Every EOA and contract accounts have an address which is used to carry out transactions. The EOA is controlled by a private key generated by the users. Anyone with an EOA can receive transactions and can send transactions if there is enough ether in their account. In order to send a transaction from an EAO the transaction must be signed using the private key. On the other hand, a contract account has ether and code. When a contract account receives a message, the code is triggered to execute functions on the internal storage or to send a message to another account [20].

3. **Ethereum network:**

   Two Ethereum nodes can only communicate if they have the same genesis block. Currently for various usages such as development Ethereum has three types of networks. All these networks have different genesis blocks. The three types of networks are:

   - Main Network: The main net is the live network of Ethereum.
   - Test Network: This net is used for testing smart contracts and decentralized apps before they are deployed on the Main network. Currently the test networks of Ethereum are Ropsten and Rinkeby.
   - Private Network: a private network can be established by anyone to create permissioned blockchains. A new genesis block must be created in order to establish a private network.

4. **Ethereum Clients**

   An Ethereum client is a software application that implements the Ethereum specification and communicates over peer-to-peer network with other Ethereum clients [23]. In addition to using the client for sending and receiving ether, a client can also be used to deploy and watch smart contracts. Currently there are six main implementations of Ethereum clients:

   - Parity, written in Rust
   - Geth, written in Go
   - cpp-ethereum, written in C++
   - pyethereum, written in python
   - Mantis, written in Scala
   - Harmony, written in Java

5. **Ethereum gas**

   Every transaction on the Ethereum blockchain is required to cover the computation, this cost is covered by paying gas to the transaction originator [20].

6. **Ethereum virtual machine**

   The Ethereum virtual machine (EVM) is the part of Ethereum that handles smart contract deployment and execution. An a high level the EVM can be seen as a global decentralized containing millions of executable objects, each with its own permanent data store [23].

7. **Ethereum block**

   The Ethereum block contains the block header and list of transactions.

8. **Messages**

   The message is the data and the value that is passed between two accounts.

9. **Ethash**

   Ethash is the Proof of Work (PoW) algorithm used in Ethereum. It is the latest version of the Dagger–Hashimoto algorithm [20].

7.5 Implementation of proof-of concept

The proof of concept application that has been developed for this thesis is mainly based on smart contracts. The smart contract is written in solidity. Solidity is a contract-oriented programming language for writing smart contracts on Ethereum. By syntax solidity is very similar to JavaScript and C.

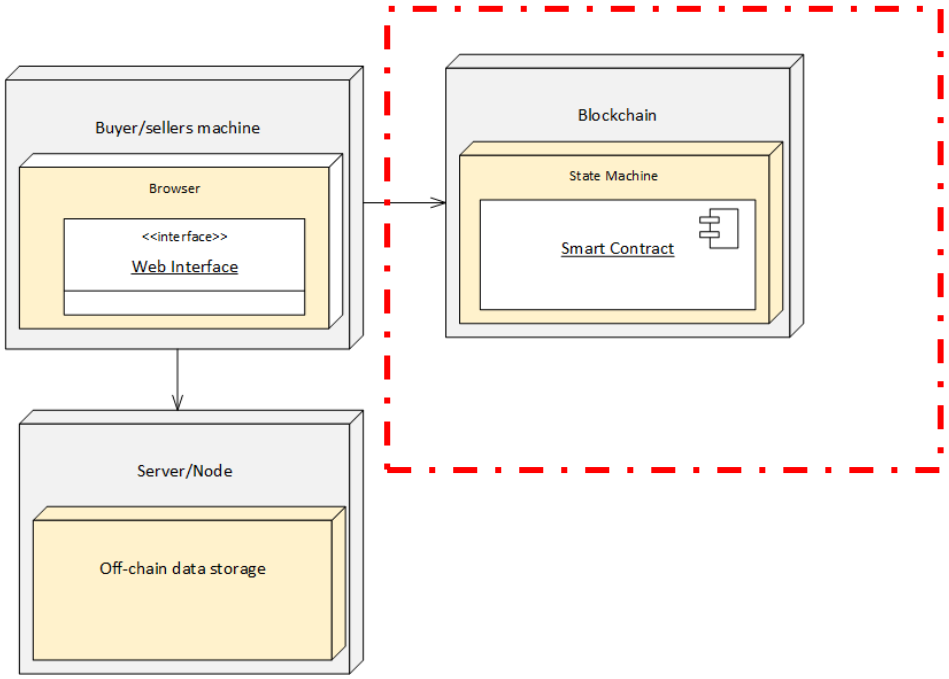

Figure 24 Deployment view, the red frame shows the part of the proof-of-concept implementation

(Source: own analysis)

Section 6.3 shows the physical view for the data marketplace proposed in this thesis. The application front-end communicates with the blockchain in a blockchain-based decentralized data marketplace. For the proof-of-concept implementation, the smart contract which contains the business logic is tested and deployed. The smart contract implantation shows the capabilities that can be implemented in a decentralized application, without the need of central authorities.

The smart contract is deployed and tested on Mist. Mist is the official Ethereum wallet which lets users create, deploy and use smart contracts [18]. The Mist client has various features such as being able to create Ethereum accounts and connecting to the various networks on Ethereum.

Figure 22 shows a proposed designed architecture for the data marketplace where the web application communicates with the smart contracts running on blockchain. For the proof-of--concept implementation only the smart contract is deployed and tested. Once the business logic and smart contracts are developed a web application can be created through which users can interact with the data marketplace.

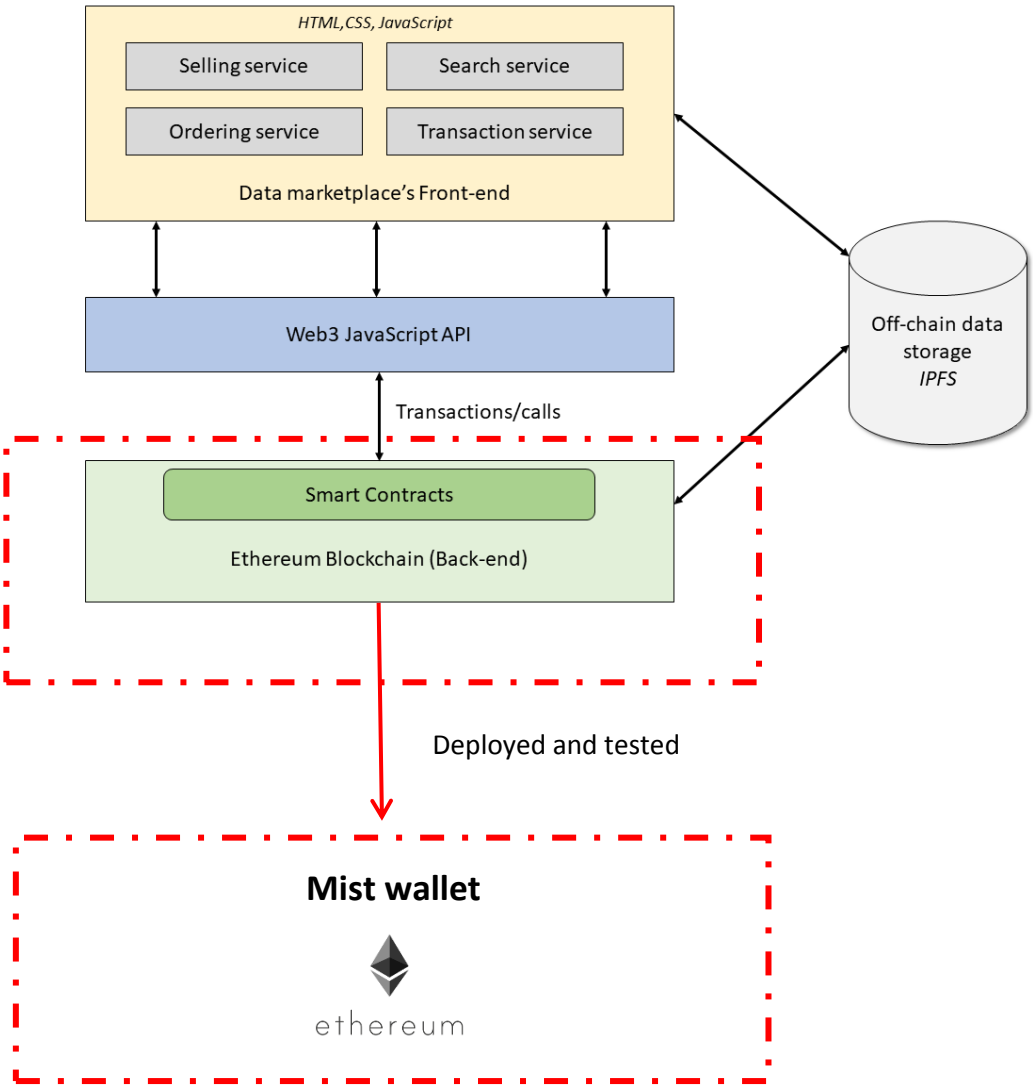

Figure 25 Design architecture, the red frame shows the part of the proof-of-concept implementation

(Source: own analysis)

Figure 24, shows the part of the design architecture for which the proof-of-concept implementation is done.

The proof-of-concept smart contract is deployed on a private network. The private network runs using Geth. Geth is the go language implementation of the Ethereum specifications and is also known the official implementation of the Ethereum client [23].

The smart contract is implemented for the buying process shown in Figure 18. When the buyer initiates the buying request, the smart contract checks whether the buyer has enough funds in his/her account for buying the requested offer. If the buyer has enough fund, the ownership of the dataset is transferred from the seller to the buyer and the amount of the order is deducted from the buyers account and sent to the sellers account.

Smart contract for buying data

```
1. pragma solidity ^0.4.0;
2.
3. // the contract to buy data
4.
5. contract Buydata {
6.     address public owner;
7.
8.     //the data can only be bought if its already not sold
9.     bool public sold = false;
10.    string public dataDescription = 'Demo data set';
11.     string private ipfsHash = '0x7B502C3A1F48C8609AE212CDFB639DEE39673F5E'
   ;
12.
13.    uint price = 10 ether;
14.
15.    function SalesContract()public payable {
16.        owner = msg.sender;
17.    }
18.
19.    // check if payment
20.    function buy() public payable {
21.        if(msg.value >= price) {
22.            owner.transfer(address(this).balance);
23.            owner = msg.sender;
24.            sold = true;
25.        } else {
26.            revert();
27.        }
28.    }
29. }
```

7.6 Evaluation

Based on the proof-of-concept implementation of the smart contract for buying data, it can be evaluated that a data marketplace based on blockchain and smart contracts can carry out transaction and ownership transfer without the need of third parties thus enabling more direct communication between the sellers and buyers.

8. Conclusion

8.1 Challenges and Limitations.

Although blockchain is a revolutionizing technology enabling more decentralized applications, while designing the system architecture, developing the proof of concept and evaluating the technologies for the decentralized data marketplace there are various challenges and limitations identified. As with any other technology these challenges need to be addressed in order to make it more robust and useful. These challenges serve as a basis for the future work.

In this thesis the challenges are categorized into two types:

1. Technical challenges and limitations:

- **Standardization:** Blockchain technology is not yet mature enough to be able to readily integrate with existing systems. Even, as the current technology stands, two blockchain networks cannot easily talk to each other [13]. Like any other evolving and developing technology, it so sometimes not stable and introduces major changes and upgrades as more requirements and bugs are identified. But these changes slower the development process on the top of these technologies. Standardization will help to improve interoperability, adoptability, and integration aspects of blockchain technology.

- **Scalability:** Scalability is one important challenge limiting blockchains for wider acceptance. Bitcoin, in its current form, can handle around 7 transactions per second (TPS). Ethereum on the other hand can manage to run up to 20 smart contracts [18]. But this is still unacceptably low throughputs for most business applications.

- **Privacy:** Privacy of transactions is a much-desired property of blockchains. However, due to its very nature, especially in public blockchains, everything is transparent [13]. Alternative solutions such as private and consortium aims to tackle this problem, but more platform stability and development for these initiatives is required.

- **Security:** Even though blockchains are generally secure recently there have been some attacks such as transaction malleability, eclipse attacks are identified [13]. Thus, further research for making blockchains more secure is required. Security in smart contracts has also become very important topic of further research after the infamous DAO hack.

2. Challenges with data trading:

Data as a commodity sold online, seems like any other product sold online. But data is quite different than other products sold online. Following are the identified challenges with respect to data trading.

- **Real-time and non-real time data requirement:**
  Although there are many types of data, in terms of data trading the data can be classified whether it is sold in real time or non-real time. Based on whether the data is required in real time or non-real time the architecture of the data marketplace changes. In system architecture shown is this thesis only focuses on real time data but further research in real-time data services is required.

- **Establishment of data integrity:**
  Ensuring that the source of the data is legitimate and has not been modified [32]. As data can be easily modified and tampered, the guarantee that the data is not modified is required. Also, fake data can be generated thus there should be a mechanism for showing data integrity.

- **Ensuring data quality:**

  The buyer should be able to check the quality of data before buying it [32]. When a user buys a physical product online, this product can be returned or exchanged if he/she is unsatisfied with it. But the same does not applies for datasets. Once the buyer sees the datasets I.e. buys them, it cannot be returned. Thus, one big challenge identified for trading data is ensuring the data quality.

8.1 Summary

This thesis outlines the requirements, system architecture and challenges for establishing a decentralized data marketplace. Moreover, a proof-of concept smart contract for the functionality for buying data is shown. The proof-of-concept smart contract shows the capability of smart contracts to carry out transaction without the need of a third party. This gives a basis and insights into the possibilities of smart contracts working on blockchain thus leading towards decentralized applications and enabling direct transactions between the sellers and buyers. The current blockchain technologies can be used to implement a decentralized data marketplace for achieving some requirements such as transaction validation and ownership transfer, but there are various challenges and technological evolvement required in order to establish the system architecture of a data marketplace proposed in this thesis.

8.2 Future work

Section 8.1 presents various identified challenges in this thesis. These challenges serve as a basis for the future work. In this thesis it has been also identified that there are many types of challenges and complex requirements for a decentralized data marketplace. Many challenges with blockchain are because it's an evolving technology and still have some technical limitations. Microservices - also known as the microservice architecture - is an architectural style that structures an application as a collection of services that are loosely coupled and independent. The microservice architecture enables the continuous delivery/deployment of large, complex applications. It also enables developing and evolving the technology stack.

Thus, the future work will also include research in microservices with regards to decentralized data marketplaces.

Section 7. Presents a design architecture for a decentralized data marketplace. This architecture also serves as a basis for implementation of a decentralized data marketplace in the future.

9. References

[1] Arent van 't Spijker, *The New Oil: Using Innovative Business Models to turn Data Into Profit*. 2014.

[2] M. Janssen, Y. Charalabidis, and A. Zuiderwijk, "Benefits, Adoption Barriers and Myths of Open Data and Open Government," *Inf. Syst. Manag.*, vol. 29, no. 4, pp. 258–268, 2012.

[3] E. X. In and S. E. D. Ata, "Business Intelligence and analytics: From Big Data to Big Impact," vol. 36, no. 4, pp. 1165–1188, 2018.

[4] P. Koutroumpis, A. Leiponen, and L. Thomas, "The (Unfulfilled) Potential of Data Marketplaces," *ETLA Work. Pap.*, vol. 2420, no. 53, 2017.

[5] Federal Trade Commission, "Data brokers: A call for transparency and accountability," *Data Brokers Need Transpar. Account.*, no. May, pp. 1–101, 2014.

[6] R. Hannaert, "Designing a Context-aware Decentralized Marketplace for Sensor Data," 2018.

[7] M. Travizano, C. Sarraute, G. Ajzenman, and M. Minnoni, "Wibson: A Decentralized Data Marketplace," pp. 1–6, 2018.

[8] C. Knieke, S. Lawrenz, M. Fröhling, D. Goldmann, and A. Rausch, "Predictive and flexible Circular Economy approaches for highly integrated products and their materials as given in E-Mobility and ICT," *Circ. Econ. Mater. Components E-mobility – CEM²*, no. April, 2018.

[9] Ken Peffers, Tuure Tuunanen, Marcus A. Rothenberger, and Samir Chatterjee, "A Design Science Research Methodology for Information Systems Research," *J. Manag. Inf. Syst.*, vol. 24, no. 3, pp. 45–77, 2007.

[10] P. Fuller and G. McNulty, "The design and engineering challenges of a water-abrasive cutting and misting system for fire suppression," *BHR Gr. - 22nd Int. Conf. Water Jet. 2014*, pp. 277–292, 2014.

[11] M. Salminen, “A METADATA MODEL FOR HYBRID DATA PRODUCTS ON A MULTILATERAL DATA MARKETPLACE,” 2018.

[12] P. Kruchten, “Architectural {{Blueprints}} {\textbackslash}textemdash{} {{The}} ``4 + 1’’ {{View Model}} of {{Software Architecture}},” vol. 12, no. November, pp. 42–50, 1995.

[13] I. Bashir, *Mastering Blockchain*. 2017.

[14] H. Ghosh, T. Consultancy, and S. Limited, “Data marketplace as a platform for sharing scientific data,” vol. 38, no. March, 2018.

[15] F. Stahl, F. Schomm, G. Vossen, and L. Vomfell, “A classification framework for data marketplaces,” *Vietnam J. Comput. Sci.*, vol. 3, no. 3, pp. 137–143, 2016.

[16] B. F. Schmid and M. A. Lindemann, “Elements of a reference model for electronic markets,” pp. 193–201, 2002.

[17] F. Stahl, F. Schomm, L. Vomfell, and G. Vossen, “Marketplaces for Digital Data: Quo Vadis?,” *Comput. Inf. Sci.*, vol. 10, no. 4, p. 22, 2017.

[18] D. Yaga, P. Mell, N. Roby, and K. Scarfone, “Blockchain technology overview,” 2018.

[19] Satoshi Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” pp. 1–9, 2016.

[20] B. Hill, S. Chopra, and P. Valencourt, *Blockchain Quick Reference*. Packt Publishing Ltd.

[21] S. Gilbert and N. Lynch, “P51-Gilbert,” pp. 51–59, 2005.

[22] I. Bashir, *Mastering Blockchain Second Edition*. .

[23] A. M. Antonopoulos and Dr.Gavin wood, *Mastering Ethereum*, First Edit. O’Reilly Media, Inc.

[24] M. Swan, *Blockchain: Blueprint for a New Economy*. 2015.

[25] Nick Szabo, “Formalizing and securing Relationships on public networks,” *https://journals.uic.edu*, 1997.

[26] J. Eberhardt and S. Tai, “On or Off the Blockchain,” *Insights Off-Chaining Comput. Data*, 2017.

[27] Storj Labs, “Storj : A Decentralized Cloud Storage Network Framework,” pp. 1–90, 2018.

[28] Streamr, “Unstoppable Data for Unstoppable Apps : DATAcoin by Streamr,” pp. 0–27, 2017.

[29] Roger Haenni, “Datum White Paper,” 2017.

[30] J. O. Grady, *System Requirements Analysis (e-bok)*, vol. 1. 2006.

[31] Ralph R. Young, *The Requirements Engineering Handbook*. ARTECH HOUSE, INC., 2004.

[32] S. Lawrenz, P. Sharma, and A. Rausch, “Blockchain Technology as an Approach for Data Marketplaces,” *Icbct 2019*, no. March, 2019.